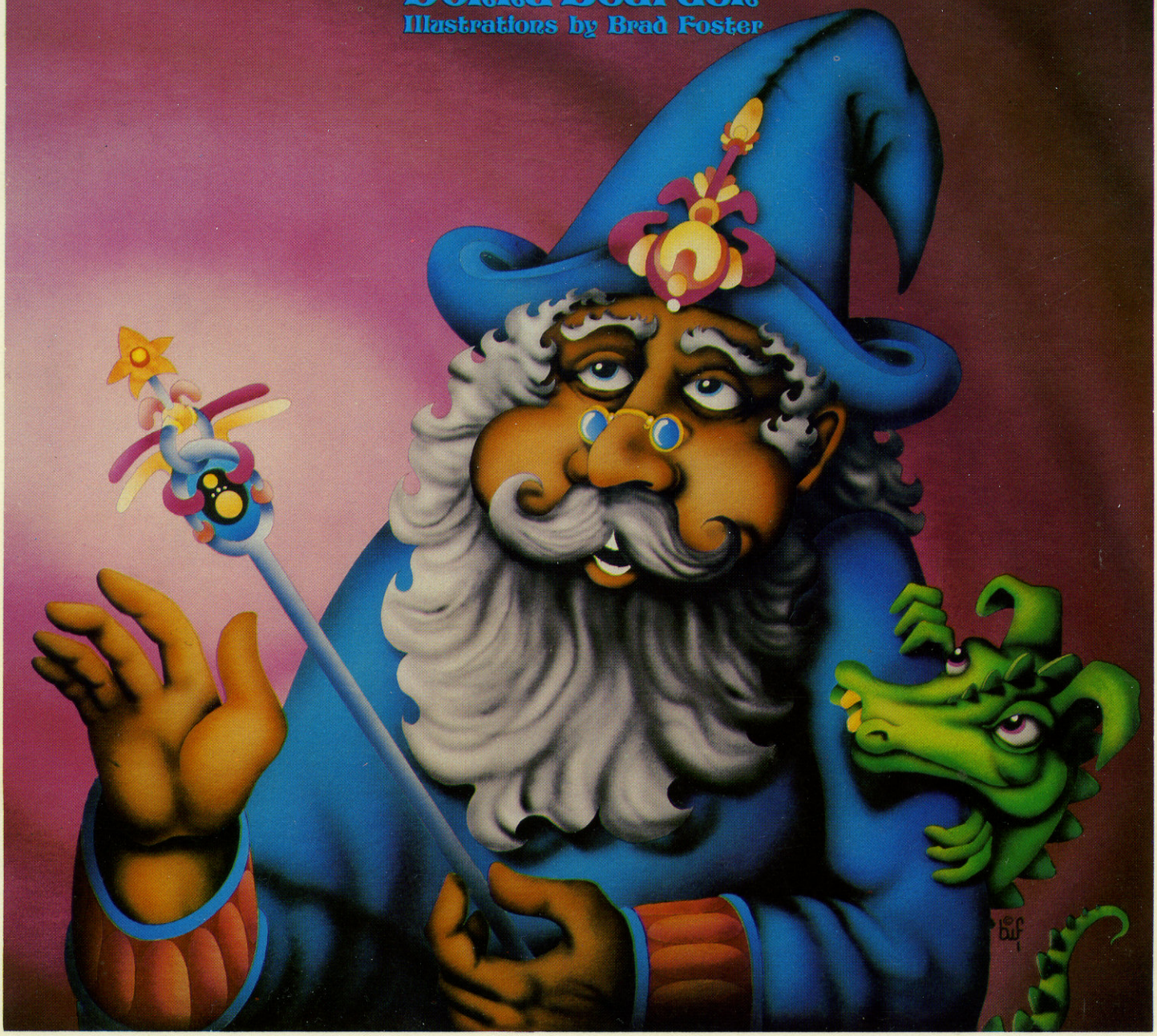
TI LOGO
TI-99/4A®

A Bit of Logo Magic

Adventures for Intermediate Programmers

Donna Bearden
Illustrations by Brad Foster



A Bit of LOGO Magic

A Bit of LOGO Magic

for the TI 99/4A

by Donna Bearden

Reston Publishing Company, Inc.
Reston, Virginia
A Prentice-Hall Company

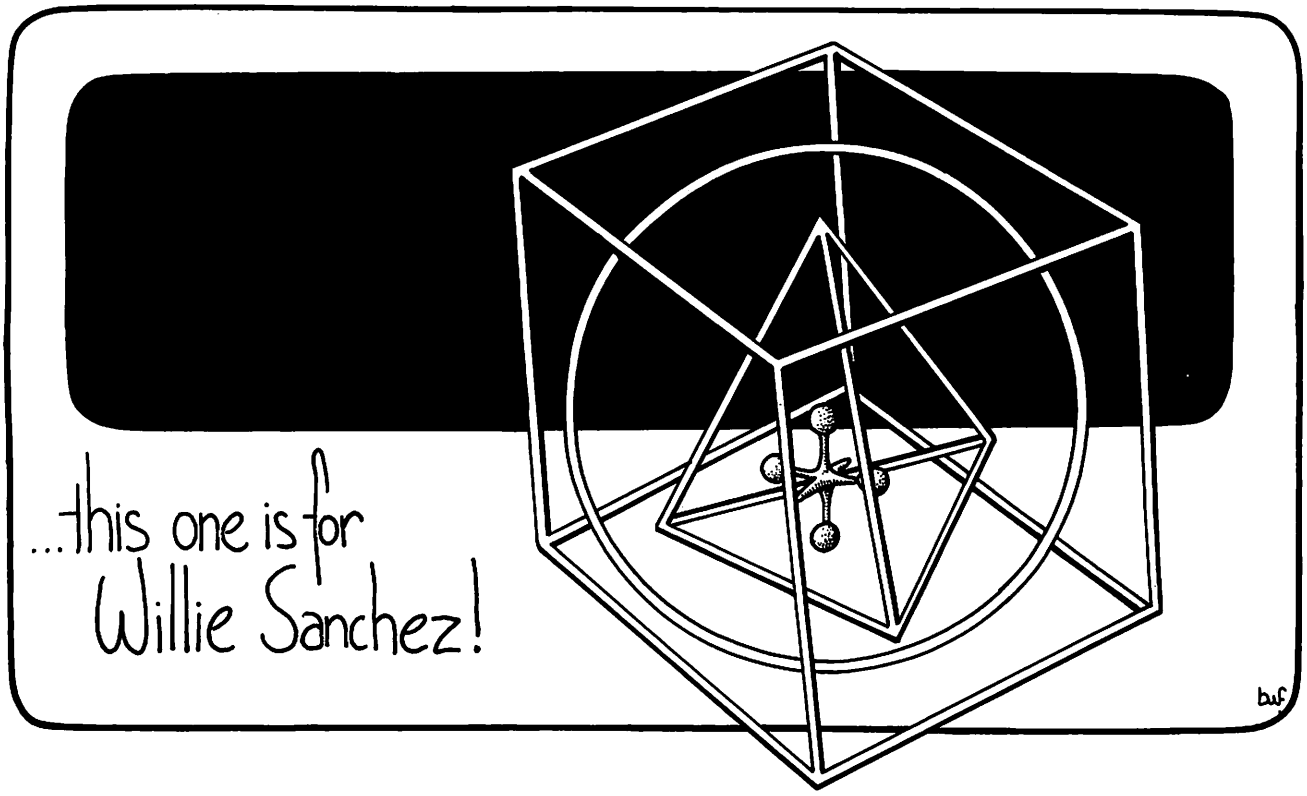
ISBN 0-8359-0494-6

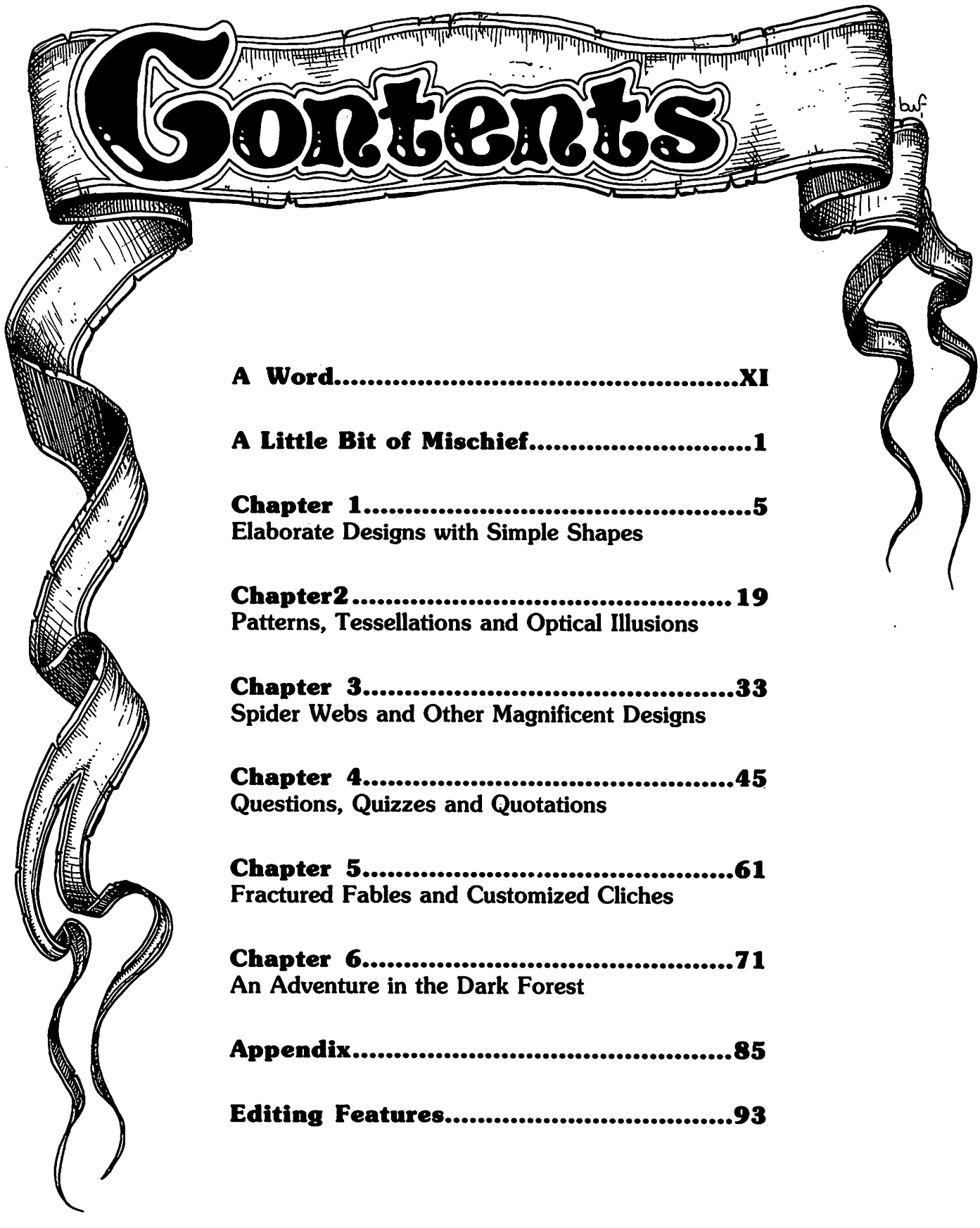
Text © 1984 by Donna Bearden
Illustrations © 1984 by Brad W. Foster
Published 1984 by Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia 22090

All rights reserved. No part of this book may be reproduced in any way, or by any means, without permission in writing from the publisher.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America


...this one is for
Willie Sanchez!
buf



A Word.....XI

A Little Bit of Mischief.....1

Chapter 1.....5
Elaborate Designs with Simple Shapes

Chapter2.....19
Patterns, Tessellations and Optical Illusions

Chapter 3.....33
Spider Webs and Other Magnificent Designs

Chapter 4.....45
Questions, Quizzes and Quotations

Chapter 5.....61
Fractured Fables and Customized Cliches

Chapter 6.....71
An Adventure in the Dark Forest

Appendix.....85

Editing Features.....93

Little Bit helped design this book especially for:

_____

If you're just starting out in LOGO, you might want to take a look at *1, 2, 3 My Computer and Me.* If you are familiar with the basic commands of LOGO, can use the edit mode, and are ready to move on past the easy stuff, then *A Bit of Logo Magic* is the book for you.

Aristotle, the wise old wizard, and Little Bit, the mischievous dragon, will give you some ideas throughout this book. But the rest is up to you. You're in charge here. There's lots of room to experiment and that's really the way you learn. So have fun! Play around! Don't be afraid to try some crazy things. Sometimes something quite unexpected will happen and you'll be off on a new adventure!

The procedures in this book were written for the TI 99/4 and TI 99/4A computers. You may use TI LOGO or TI LOGO II.

It wasn't a bad idea. In fact, most days Aristotle didn't regret it at all. But there were other times he wished he hadn't been so eager to share his experiments with a companion — or at least not one who turned out to be half mischief. He was thinking of Little Bit, of course, but maybe we better start at the beginning.

It was a gloomy day in the old castle at Cyber-by-the-Sea. Up in a room at the top of the tallest turret, Aristotle absent-mindedly twirled his moustache around one finger as he puzzled over a particularly perplexing procedure he was developing.

Unlike his ancestors from the middle ages, Aristotle was a thoroughly modern wizard. He turned more and more to his computer and used his magic staff only to put the finishing touches on certain projects. All his magic books were recorded on floppy disks. Ancient codes and languages had been replaced by LOGO.

The gloom of the day settled thickly around the old castle and Aristotle realized he would have to do something quite extraordinary to break the mood. He studied the shapes on the monitor in front of him. Suddenly he began to chuckle to himself as an idea began to form in the far reaches of his mind. He laughed out loud as the idea took shape.

"A dragon!" he said out loud.

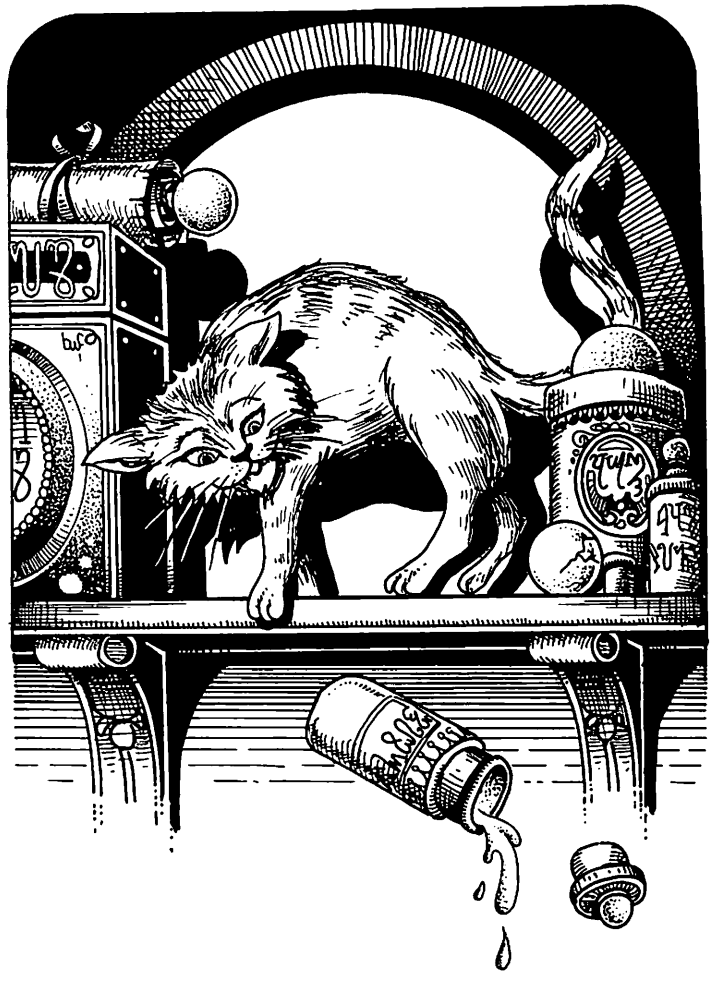
Dragons had been done away with in the middle ages, all but a few scarce ones here and there who were careful to stay hidden.

"What a crazy-wonderful idea," he answered himself. (Aristotle always held conversations with himself when he got truly excited.)

"Not a ferocious fire-breather, but a small, playful creature. One who could sit on my shoulder or on top of the monitor. A smart little fellow who might even enjoy learning LOGO."

And Aristotle whistled as he loaded the disk labeled DRAGONS. He hummed as he read the ingredients for conjuring up a "Dragon, small, friendly." He sang out loud as he gathered his spices, herbs and potions.

If he hadn't been so intent on his project, he might have noticed The Cat ("T.C." for short). T.C. was a curious feline who followed Aristotle home one day and simply refused to leave. He was quiet, usually stayed out of the way, and caught a mouse every now and then.



But the sound of the singing and the smell of the spell was too much for T.C. He leapt up on the shelf to get a better view of what was going on. In doing so, he tipped over a bottle of mischief and the lid came off. Before Aristotle could get it back on, a few potent drops fell right into the brew which was already taking the shape of a "Dragon, small, friendly."

"Oh well, no harm done," Aristotle thought to himself. "Everyone can stand a little mischief."

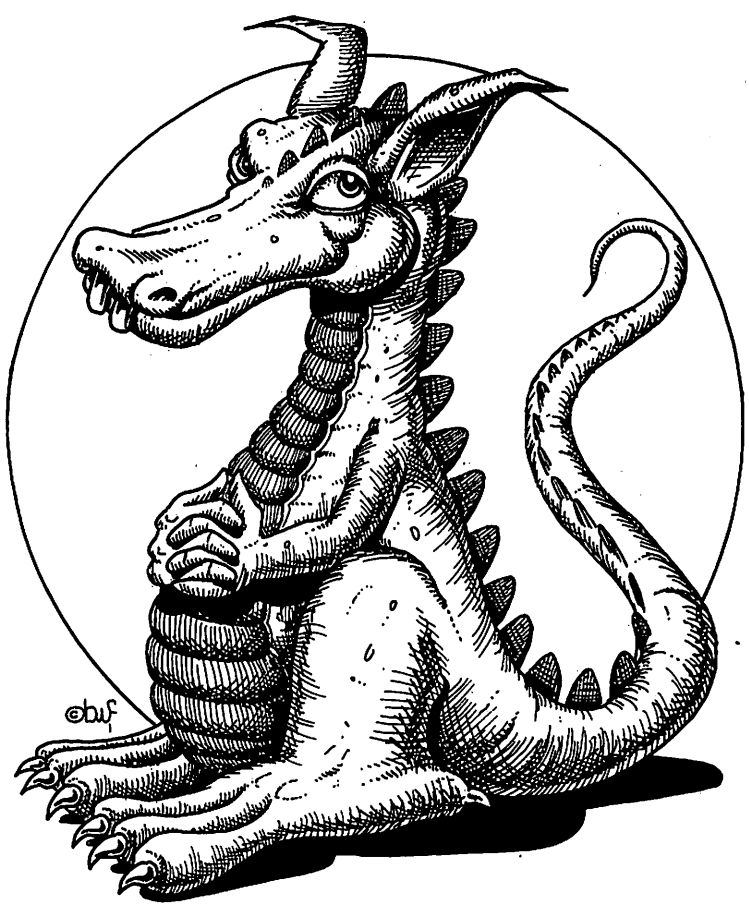
(But little did he know what was in store for him.)

Since the little dragon was made from a little bit of this and a little bit of that (and a little bit of mischief, thanks to The Cat), it seemed quite appropriate to name him "Little Bit."

Little Bit became Aristotle's constant companion. He learned very quickly, perhaps too quickly, because the more he learned, the more he delighted in altering the wizard's procedures.

Aristotle, wise wizard that he is, had planned to lead you though this book by himself. But Little Bit had ideas of his own. He sneaked in and changed commands in some of the procedures and caused some unexpected things to happen. Sometimes the wizard enjoyed the changes because he discovered something new. Other times, well, he wondered if the cirsus might be interested in a "Dragon, small, friendly, and mischievous."

Or maybe some kid out there who likes LOGO just might be interested in a Little Bit of mischief...



©brad w foster·1983

Elaborate Designs with Simple Shapes 1

Do you know how to make an equilateral triangle, a square, a pentagon, a hexagon and an octagon? How about a circle and an oval? How about stars? Can you write a procedure to draw a rectangle? If you can draw all of these shapes, you can develop some pretty elaborate drawings on the computer.

First, let's do a quick review. Fill in the missing numbers in the following procedures:

```
TO TRIANGLE
REPEAT ___ [FD  50  RT ___]
END
```

```
TO PENTAGON
REPEAT ___ [FD  45  RT ___]
END
```

```
TO CIRCLE  :N
REPEAT 36 [FD ___ RT ___]
END
```

```
TO OCTAGON  :N
REPEAT ___ [FD  :N RT ___]
END
```

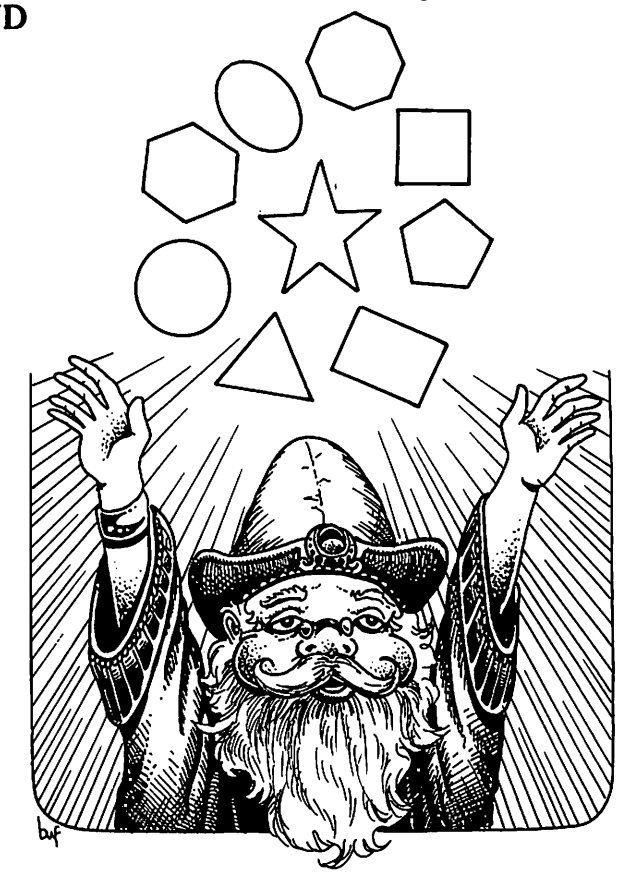
```
TO OVAL  :N  :S
REPEAT 18 [FD  :N RT  10]
FD  :S
REPEAT 18 [FD  :N RT ___]
FD ___
END
```

```
TO STAR  :N
REPEAT 5 [FD  :N RT _____]
END
```

```
TO SQUARE
REPEAT ___ [FD  35  RT ___]
END
```

```
TO HEXAGON
REPEAT ___ [FD ___ RT _____]
END
```

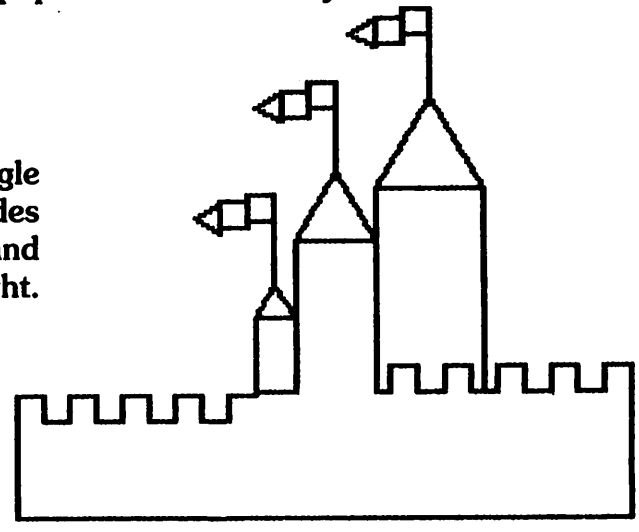
```
TO CIRCLE  :N
REPEAT 72 [FD ___ RT ___]
END
```



Now let's start playing with some of these shapes and see what we can do. Anyone who starts working with LOGO puts a triangle on top of a square to make a house. You might have added a couple of windows, a chimney, a door, and perhaps even a picket fence out front. That was a pretty good start.

But wizards and dragons don't live in houses with picket fences. They live in castles. Can you design a castle using the shapes you already know how to make on the computer? Try a couple of designs on paper first and then try to do one in LOGO.

Here's a simple one to start with. You can add more details to it.



Put a couple squares and a triangle together to make a flag. This flag includes a square with the Turtle turning left and another one with the Turtle turning right.

```
TO FLAG
FD 15
SQL 8
LT 90 FD 8 RT 90 BK 2 LT 90
SQ 8
FD 8 LT 90 BK 7
TRI 9
END
```

```
TO TRI  :N
REPEAT 3 [FD  :N RT 120]
END
```

```
TO SQ  :N
REPEAT 4 [FD  :N RT 90]
END
```

```
TO SQL  :N
REPEAT 4 [FD  :N LT 90]
END
```

To make a spire, first define a rectangle. If you make one side twice as long as the other, you can make one side :X and the other side :X times 2. In LOGO, it looks like this: X * 2. Here's the rectangle procedure:

```
TO REC  :X
REPEAT 2 [FD  :X * 2 RT 90 FD  :X RT 90]
END
```

Put a triangle on top of the rectangle and the flag on top of the triangle. By using variables throughout the procedure, you can make any size spire you want.

```
TO SPIRE  :X
REC  :X
FD  :X * 2 RT 30
TRI  :X
FD  :X LT 30
FLAG
END
```

The wall is made of half-squares. Define two half-square procedures, one with the Turtle turning right and one with the Turtle turning left:

```
TO H.SQ
REPEAT 2 [RT 90 FD 8]
END
```

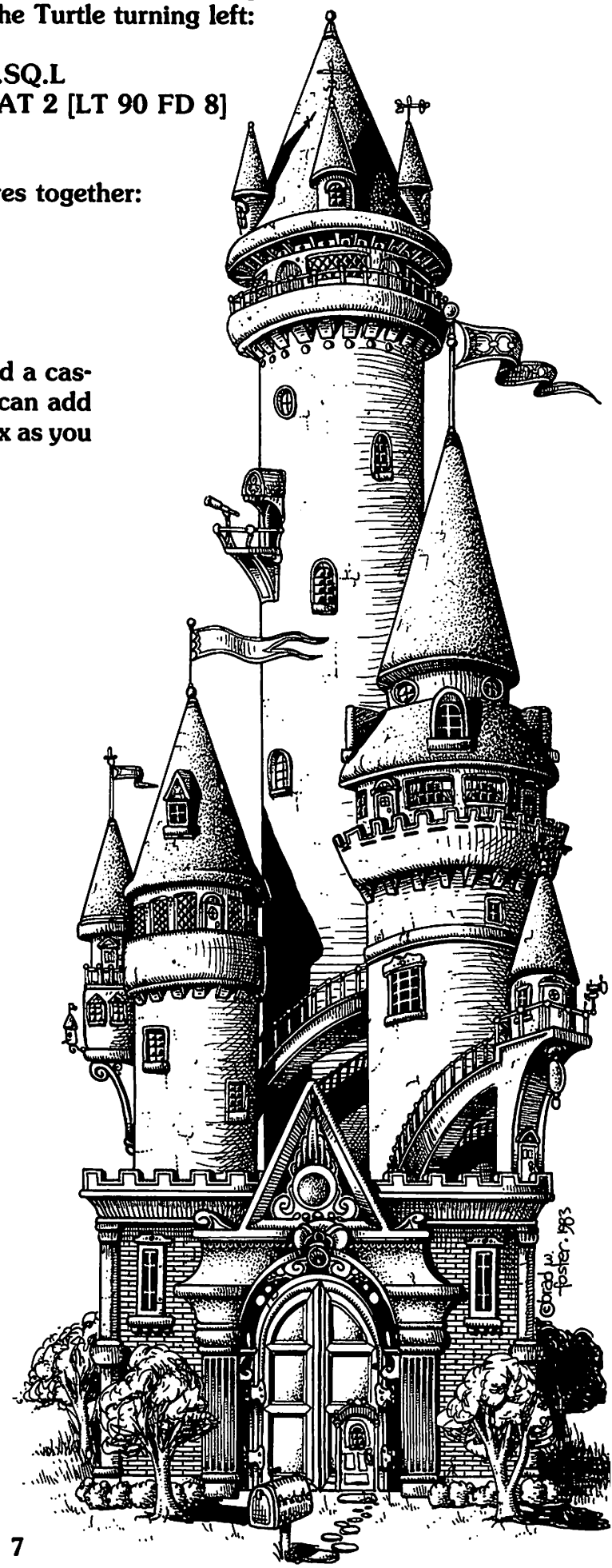
```
TO H.SQ.L
REPEAT 2 [LT 90 FD 8]
END
```

To make a wall, put some half-squares together:

```
TO WALL
REPEAT 4 [H.SQ.L  H.SQ]
END
```

Now we have all the elements to build a castle. This one has three spires, but you can add more. In fact, you can make it as complex as you want.

```
TO CASTLE
TELL TURTLE HT
SY -10 SPIRE 25
MOVE1
SH 0  SPIRE 20
MOVE2
SH 0  SPIRE 10
MOVE2
SH -90  WALLS
END

TO WALLS
MOVE3
WALL  H.SQ.L
FD 27 LT 90 FD 180 LT 90 FD 45
WALL  H.SQ.L
RT 90 FD 5 PU FD 20 PD FD 15
END

TO MOVE1
SXY -20 (-10)
END

TO MOVE2
SXY -30 (-10)
END

TO MOVE3
PE BK 65 FD 65 RT 90 PD
END
```



Can you add some windows, an entrance, a moat and a drawbridge? How about putting a sun in the sky?

```
TO SUN
TELL SPRITE 1
CARRY  :BALL
SC  :YELLOW
SXY 70 90
END
```

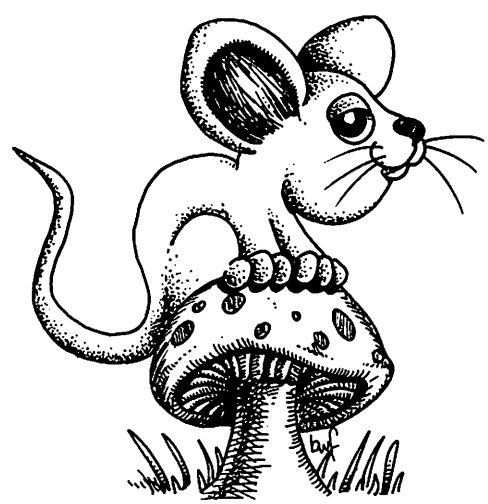
Castles and houses are mostly straight lines and angles. What about something with curves?

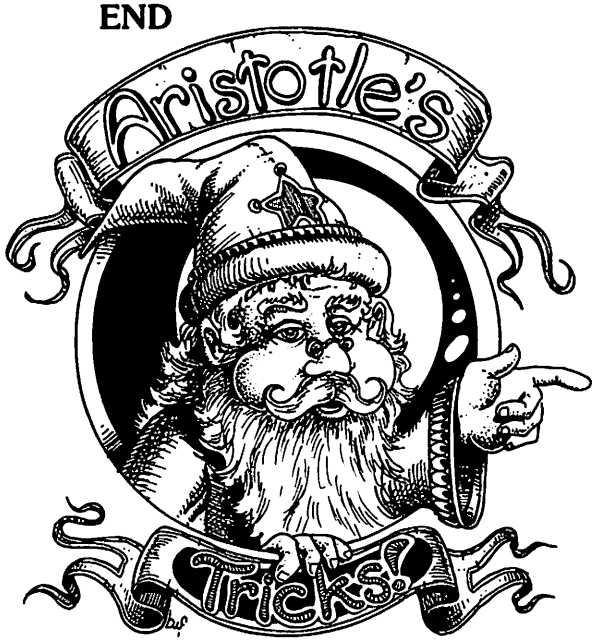
Out in Aristotle's mushroom garden, there's a little mouse named Tiny, who's always very careful to avoid The Cat.

Tiny is made from circles, an oval, a triangle, some pieces of circles, and a few straight lines. First we'll develop the general shape and then we'll go back into some of the procedures and erase a few lines and add a few details to make Tiny more real.

We'll need procedures for a circle, an oval, and a triangle:

```
TO CIR  :N
REPEAT 20 [FD  :N RT 18]
END

TO OVAL  :N  :S
REPEAT 18 [FD  :N RT 10]
FD  :S
REPEAT 18 [FD  :N RT 10]
FD  :S
END

TO TRI  :N
REPEAT 3 [FD  :N RT 120]
END
```





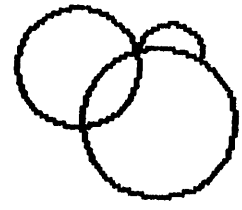
A Note from Aristotle:

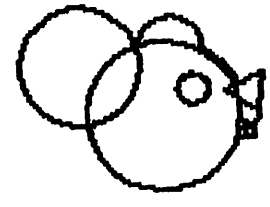
Remember, anytime you're developing a complex drawing, work in chunks. Design part of it and write a procedure. Design another part and write another procedure. It's much easier to edit if you've got several smaller procedures than if you've got one big one.

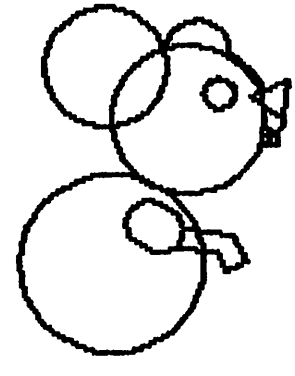
You can see that one of Tiny's ears is a half circle and the part under his nose is part of a circle. How do you make half of a circle? What about his tail? Can you figure out how it's made? Play around with pieces of circles (arcs) and see what you can do.

Here are all the procedures to draw the rough sketch of Tiny. If you type in all the procedures to draw the rough sketch and then edit them to make the final drawing, you can watch the changes take place on the screen and experience the process of refining Tiny just like we did.

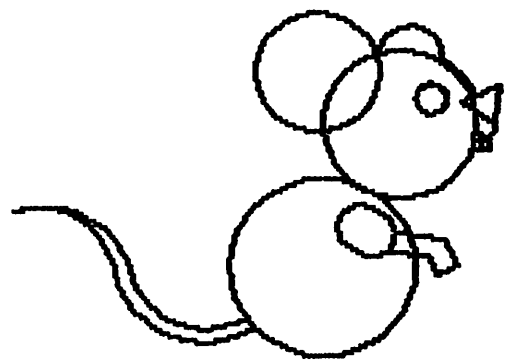
```
TO MOUSE
HEAD ADD.NOSE
ADD.TEETH
ADD.EYE
ADD.BODY
ADD.ARM TAIL
END

TO HEAD
CIR 5 LT 90 CIR 4 RT 90
REPEAT 5 [FD 5 RT 18]
LT 70
REPEAT 10 [FD 2 RT 18]
END

TO ADD.NOSE
LT 65
REPEAT 4 [FD 3 RT 10]
LT 100 BK 3 TRI 11
FD 11 RT 120 FD 11
REPEAT 7 [FD 1 RT 18]
END

TO ADD.TEETH
REPEAT 5 [LT 10 BK 1]
LT 50 FD 4 RT 90 FD 2 RT 90
FD 4 RT 180 FD 4 RT 90 FD 2 RT 90 FD 4
END

TO ADD.EYE
PU LT 100 FD 15 PD CIR 1
END

TO ADD.BODY
PU LT 80 FD 18 LT 100 PD CIR 6
END

TO ADD.ARM
PU RT 90 FD 4 LT 90 FD 7 PD
ARM
END
```







```
TO ARM
OVAL 1 3
REPEAT 6 [FD 1 RT 10]
LT 90 FD 12
RT 60 FD 8 RT 90 FD 5
RT 90 FD 5 LT 60 FD 10
END

TO TAIL
PU LT 30 FD 30 PD
REPEAT 10 [FD 3 RT 10]
REPEAT 10 [FD 4 LT 10]
LT 170
REPEAT 10 [FD 3 RT 10]
LT 10
REPEAT 11 [FD 4 LT 10]
END
```



Now the fun starts. We need to go back into the various procedures and tell the Turtle to pick up his pen so some of the lines don't show.

Look at Tiny's head. His ear would look better if the top of his head didn't show through, so we'll have to figure out how not to draw that part of the circle. Also, we could raise the pen to get rid of the extra line at his nose.

Compare these two procedures. (In the following procedures, the rough sketch procedures will be on the left and the finished drawing procedures will be on the right.)

```
TO HEAD
CIR 5



LT 90 CIR RT 90
REPEAT 4 [FD 5 RT 18]
LT 70
REPEAT 10 [FD 2 RT 18]
END
```

```
TO HEAD
PU REPEAT 5 [FD 5 RT 18] PD
REPEAT 5 [FD 5 RT 18]
PU REPEAT 2 [FD 5 RT 18] PD
REPEAT 8 [FD 5 RT 18]
LT 90 CIR 4 RT 90
PU REPEAT 5 [FD 5 RT 18] PD
LT 70
REPEAT 10 [FD 2 RT 18]
END
```

Do you notice how CIR 5 became four separate commands? That's because we want the Turtle to draw two parts of the circle and lift his pen for two other parts. We just have to play around with how many REPEATS to include in each command.

We know that the Turtle starts right at the bottom of the ear, so the first command will be with the pen up. It looks like the ear covers not quite one-fourth of the circle of the head. Since there are 20 REPEATS in our circle procedure, and one-fourth of 20 is 5, let's try PU REPEAT 5 [FD 5 RT 18] PD for the first command.

It looks about the same distance from the top of Tiny's ear, where we want the Turtle to start drawing again, to the start of his nose, where we want the Turtle to lift up his pen again. So let's try

REPEAT 5 [FD 5 RT 18]

for the second command.

From the top of his nose to the bottom of his nose looks like about one-half of the previous distance, so let's try a REPEAT 2 for it. Since we want the pen up, the command should be

PU REPEAT 2 [FD 5 RT 18] PD

How much should the last command be? To make a complete circle, the four REPEAT commands must add up to 20. So far, we have

PU REPEAT 5 [FD 5 RT 18] PD

REPEAT 5 [FD 5 RT 18]

PU REPEAT 2 [FD 5 RT 18] PD

so the last command will be

REPEAT _____ [FD 5 RT 18]

We don't need to make any changes to the ADD.NOSE and ADD.TEETH procedures.

Let's add an eyeball to the eye:

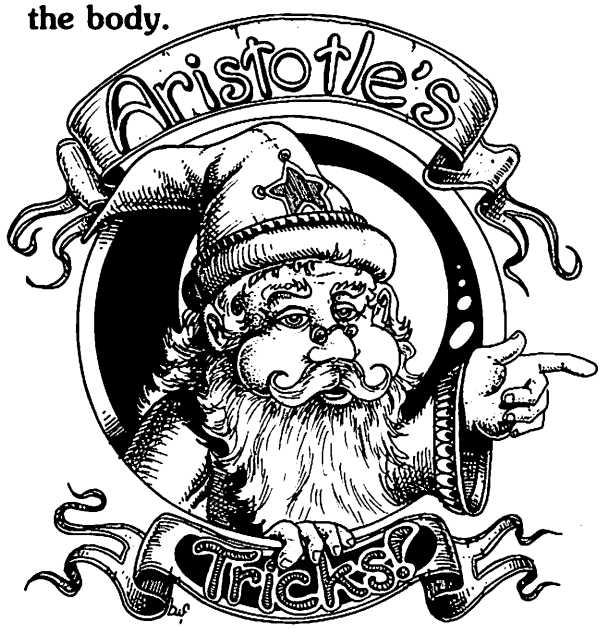
```
TO ADD.EYE
PU LT 70 FD 20 PD CIR 1
END
```

```
TO ADD.EYE
PU LT 70 FD 20 PD CIR 1
REPEAT 12 [FD 1 RT 30]
END
```

He'd sure look better with some whiskers, so let's add them and then add ADD.WHISKERS to the MOUSE procedure, between ADD.EYE and ADD.BODY.

```
TO ADD.WHISKERS
PU LT 30 BK 10 LT 35 PD
WHISKERS
PU RT 60 BK 10 LT 140 PD
WHISKERS
END

TO WHISKERS
FD 9 BK 9
LT 20 FD 12 BK 12
LT 20 FD 9 BK 9
END
```

Because the Turtle ends up at a different place when he draws whiskers than when he doesn't, we'll have to redefine the move to get him in the right position to start the body.



A Trick from Aristotle:

When you've figured out the right position once, it's kind of a drag to have to do it all over again and then not know for sure if you're in the right spot. Have the Turtle draw the head and move to the position to start drawing the body. Now type PRINT XCOR and press return. You'll see a number. That is the X coordinate. Type PRINT YCOR and you'll see another number. It is the Y coordinate. Write them down. Now type PRINT HEADING and press return. You'll see a number which tells you exactly where the Turtle is facing.

Now you know exactly where the Turtle was and what direction he was facing in your rough sketch and you can play around with commands until you get him as close to that position as possible. It doesn't have to be exact. (We could simply tell the Turtle to set his position followed by those two coordinates and set his heading to the correct heading. However, later we may want to move the mouse to a different position on the screen and that won't be possible if you tie him to a particular position or heading.)

```
TO ADD.BODY
PU LT 80 FD 18 LT 100 PD
CIR 6
END
```

```
TO ADD.BODY
MOVE2
REPEAT 2 [FD 6 RT 18]
PU REPEAT 9 [FD 6 RT 18] PD
REPEAT 9 [FD 6 RT 18]
END

TO MOVE2
PU LT 50 BK 33 RT 55 BK 2 PD
END
```

ADD.ARM remains the same, but there are several changes in ARM:

```
TO ARM
OVAL 1 3




REPEAT 6 [FD 1 RT 10]
LT 90 FD 12
RT 60 FD 8 RT 90 FD 5
RT 90 FD 5 LT 60 FD 10
END
```

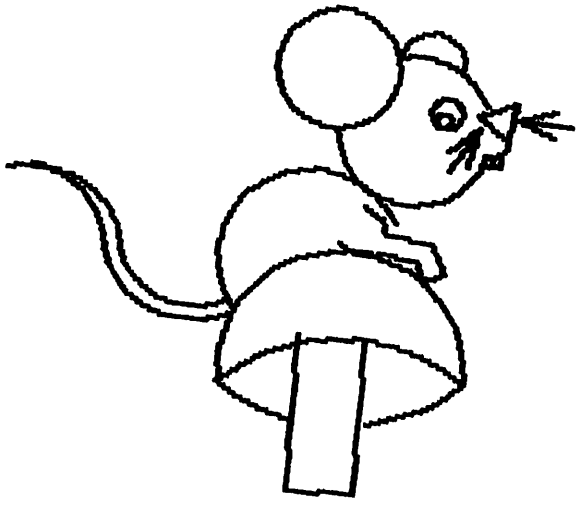
```
TO ARM
REPEAT 6 [FD 1 RT 10]
PU REPEAT 6 [FD 1 RT 10]
PD REPEAT 6 [FD 1 RT 10]
FD 3
PU REPEAT 18 [FD 1 RT 10]
PD FD 3
REPEAT 6 [FD 1 RT 10]
LT 90 FD 12
RT 60 FD 8 RT 90 FD 5
RT 90 FD 5 LT 60 FD 10
END
```

There aren't any changes in Tail.

Now let's add a mushroom.

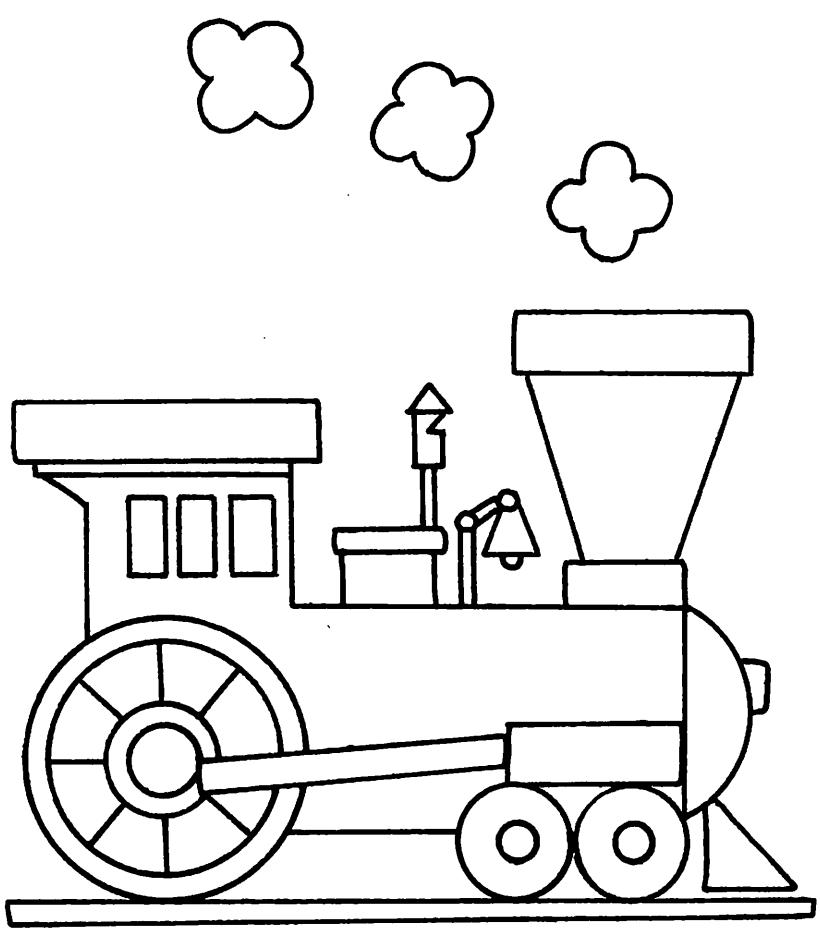
```
TO ADD.MUSHROOM
PU LT 50 BK 15 LT 10 PD
MUSHROOM
END

TO MUSHROOM
REPEAT 11 [FD 8 RT 18]
RT 115
REPEAT 7 [FD 8 LT 15]
LT 80
REPEAT 2 [FD 8 LT 15]
PU REPEAT 2 [FD 8 LT 15]
PD REPEAT 3 [FD 8 LT 15]
BK 3 LT 75
REPEAT 2 [FD 8 LT 15]
LT 90 FD 35 RT 90 FD 15 RT 90
FD 35
END
```

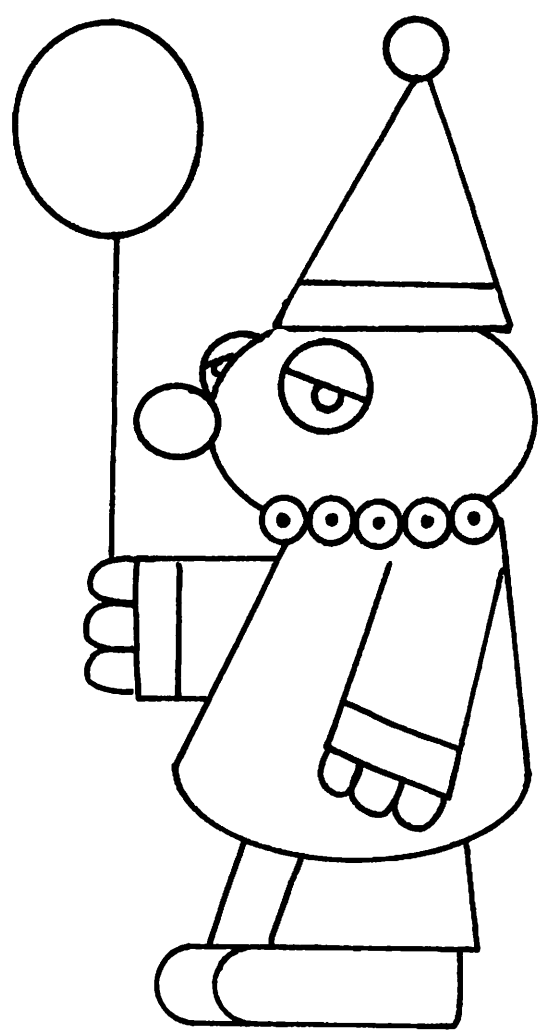
It's a good thing we didn't include a specific position within one of the procedures because we're going to have to move Tiny up higher on the screen so the mushroom will fit. Here's a START procedure and a superprocedure to draw Tiny:

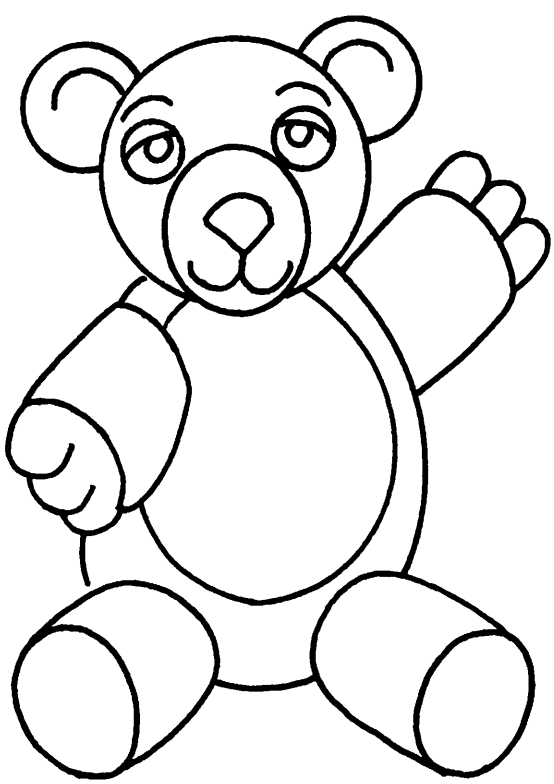


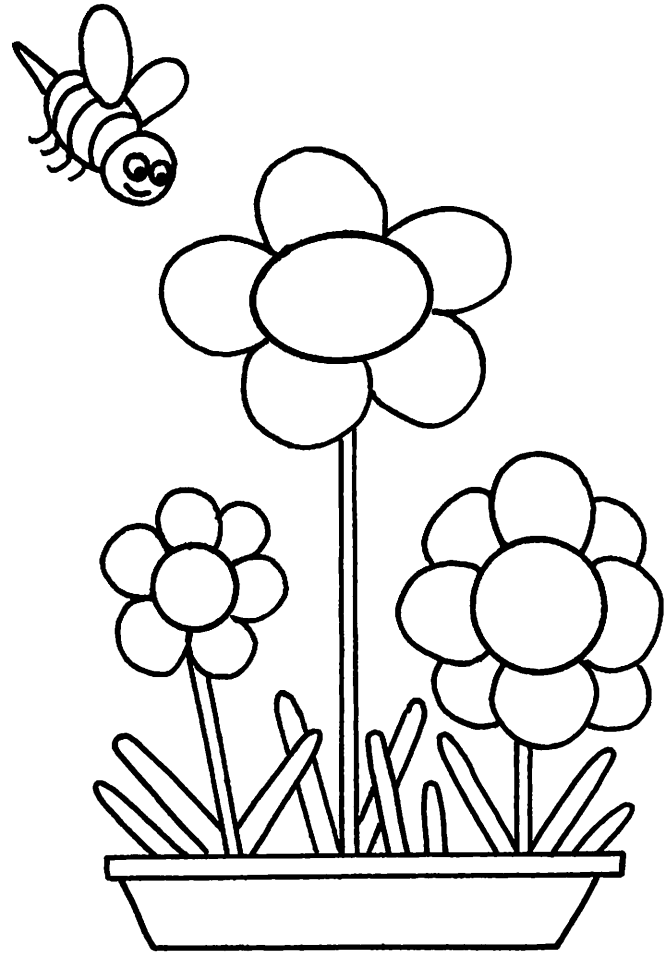
```
TO START
PU FD 40 PD
END

TO TINY
TELL TURTLE
HT
START
MOUSE
ADD.MUSHROOM
END
```

Try some of your own designs or take a crack at some of Aristotle's sketches.









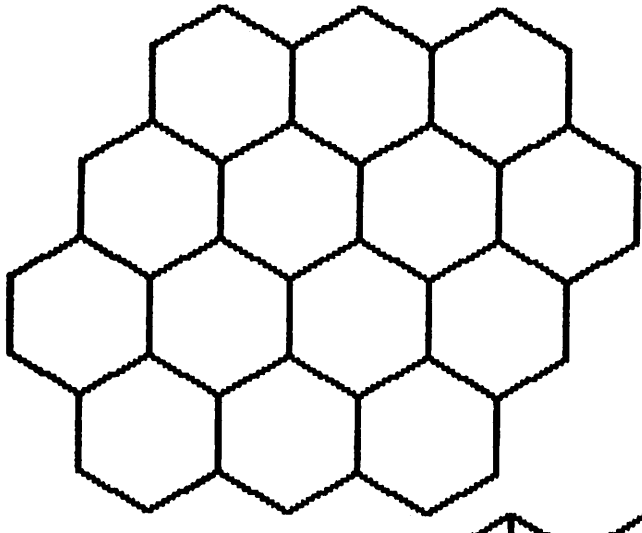
©brad w. foster 1983

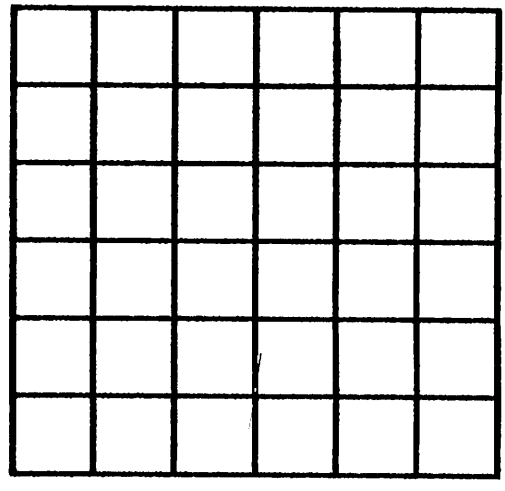
Patterns, Tessellations and Optical Illusions 2

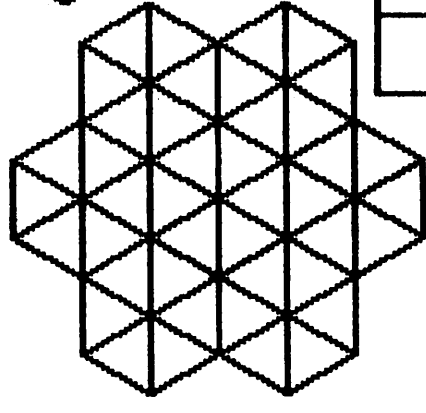
It probably all started with Aristotle's great-grandmother. From scraps of material and pieces of old wizard robes, she could perform a little magic of her own with needle and thread. Everyone who ever saw them marveled at her beautiful quilts. She seemed to have a knack for putting simple shapes together. That's probably where Aristotle developed his love for tessellations.

"Tessellation" is a very impressive word to throw around. You've seen them everywhere, but you may not have known what to call them. A tessellation is a repeating pattern of shapes, with no shapes overlapping and no gaps between the shapes. A regular tessellation is made by repeating a single regular polygon over and over to cover a plane.

If you do a little bit of experimenting with regular polygons, you'll find out there are only three shapes, a triangle, a square and a hexagon, that can be used to make a regular tessellation (made by repeating only one shape.) Can you figure out why?



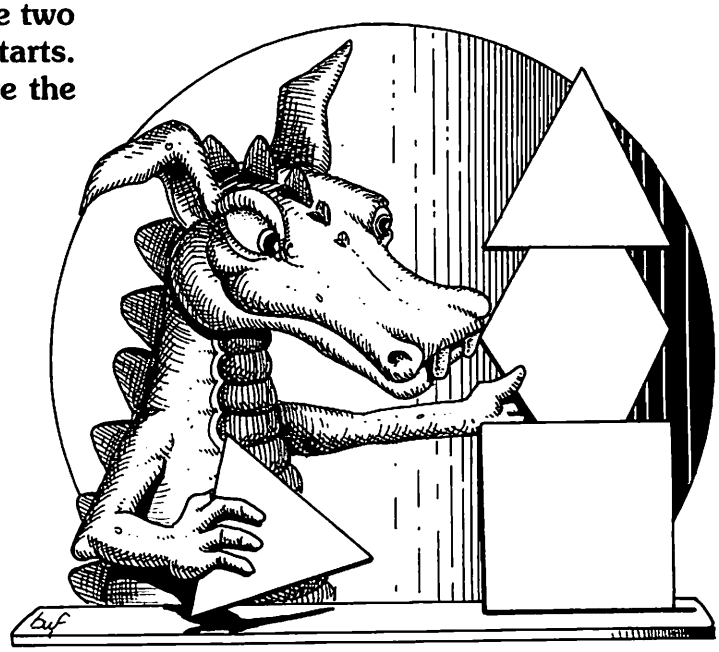


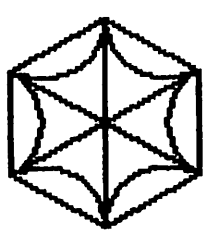


We're going to do some fancy tessellating using regular polygons. But first we'll do a simple design within the polygon. Then, as it is repeated over and over, some exciting things start happening.

Let's start with a triangle. Within the triangle, draw an arc connecting the two corners opposite where the Turtle starts. Look what happens when we rotate the triangle to make a hexagon:

```
TO TRI1
TRI 20
RT 60 FD 20 LT 175
REPEAT 11 [FD 2 RT 10]
REPEAT 11 [LT 10 BK 2]
RT 175 BK 20 LT 60
END

TO TRI  :N
REPEAT 3 [FD  :N RT 120]
END
```

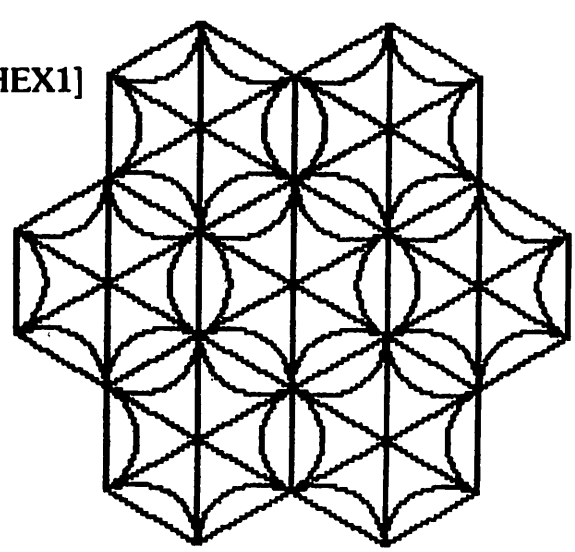




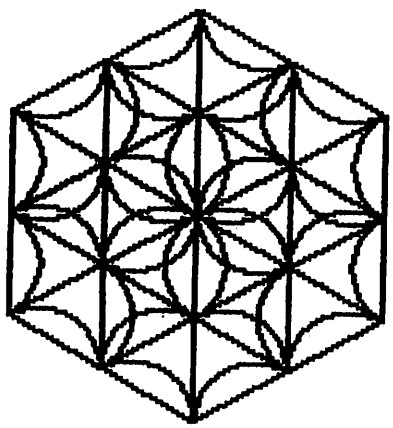
```
TO HEX1
REPEAT 6 [TRI1 RT 60]
END
```

Put several hexagons together to make a tessellation:

```
TO TRI.TESS1
PU FD 20 PD
HEX1
FD 20 RT 60 FD 20 LT 60 HEX1
REPEAT 5 [LT 120 FD 20 RT 60 FD 20 HEX1]
END
```

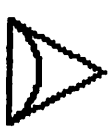


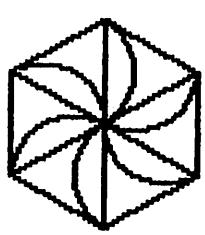
There are lots of different patterns you can put together with the simple triangle-curve design. Here's one:



```
TO ROSETTE
REPEAT 6 [HEX1 RT 60 FD 20]
END
```

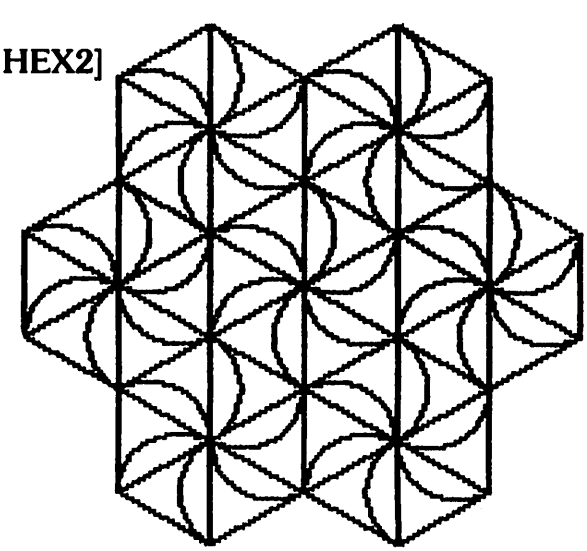
Suppose when you first defined the triangle, you put the curve on the first side the Turtle draws. When you rotate the triangle, you'll get a totally different pattern. And look what happens when you tessellate the hexagon!





```
TO TRI2
TRI 20
RT 55
REPEAT 11 [FD 2 LT 10]
REPEAT 11 [RT 10 BK 2]
LT 55
END

TO HEX2
REPEAT 6 [TRI2 RT 60]
END
```

```
TO TRI. TESS2
PU FD 20 PD
HEX2
FD 20 RT 60 FD 20 LT 60 HEX2
REPEAT 5 [LT 120 FD 20 RT 60 FD 20 HEX2]
END
```

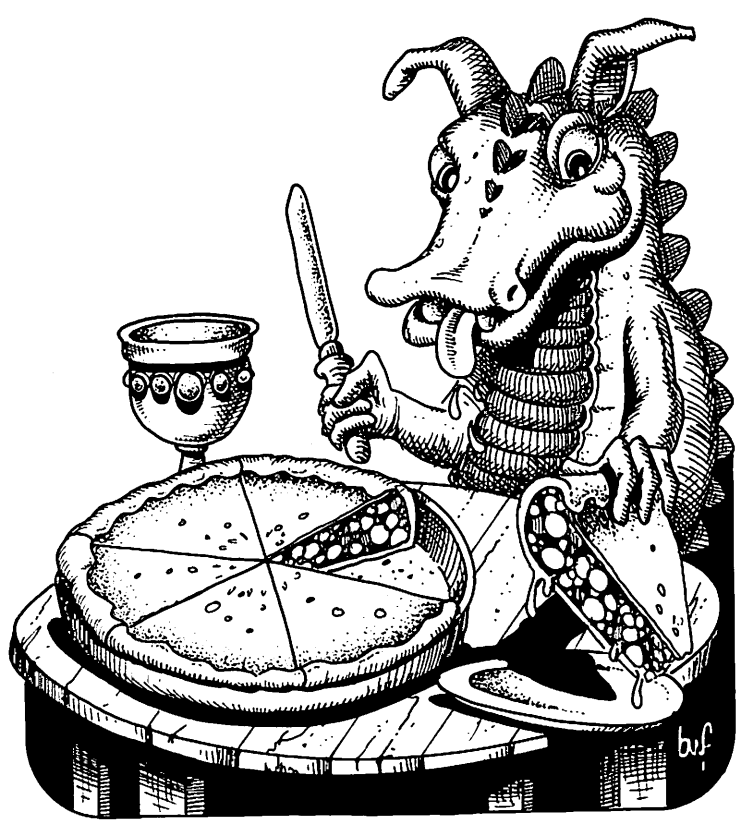


Design your own triangle tessellations.

First, draw an equilateral triangle and add some lines or curves or a shape within the triangle.

Then rotate it to make a hexagon.

And then repeat the hexagon over and over.



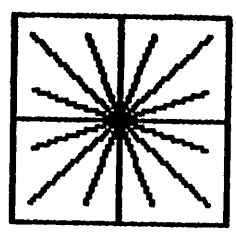
Let's try a repeating pattern with a square.

First draw a square and put a design in it. Be sure to return the Turtle to its original position.

```
TO SQ  :N
REPEAT 4 [FD  :N RT 90]
END
```

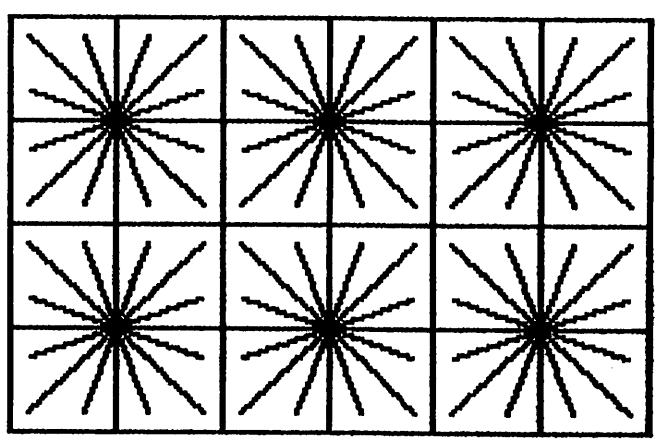
```
TO SQ1
SQ 20
RT 20 FD 17 BK 17
RT 25 FD 23 BK 23
RT 25 FD 17 BK 17
LT 70
END
```

Once you have designed a square, repeat it several times and see what happens. Here's what happens with SQ1.



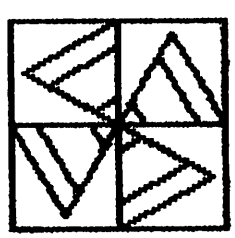
```
TO ROTATE.SQ1
REPEAT 4 [SQ1 RT 90]
END
```

```
TO QUILT
ROTATE.SQ1
REPEAT 3 [FD 40 ROTATE.SQ1 RT 90]
FD 80 ROTATE.SQ1
RT 90 FD 40 ROTATE.SQ1
END
```

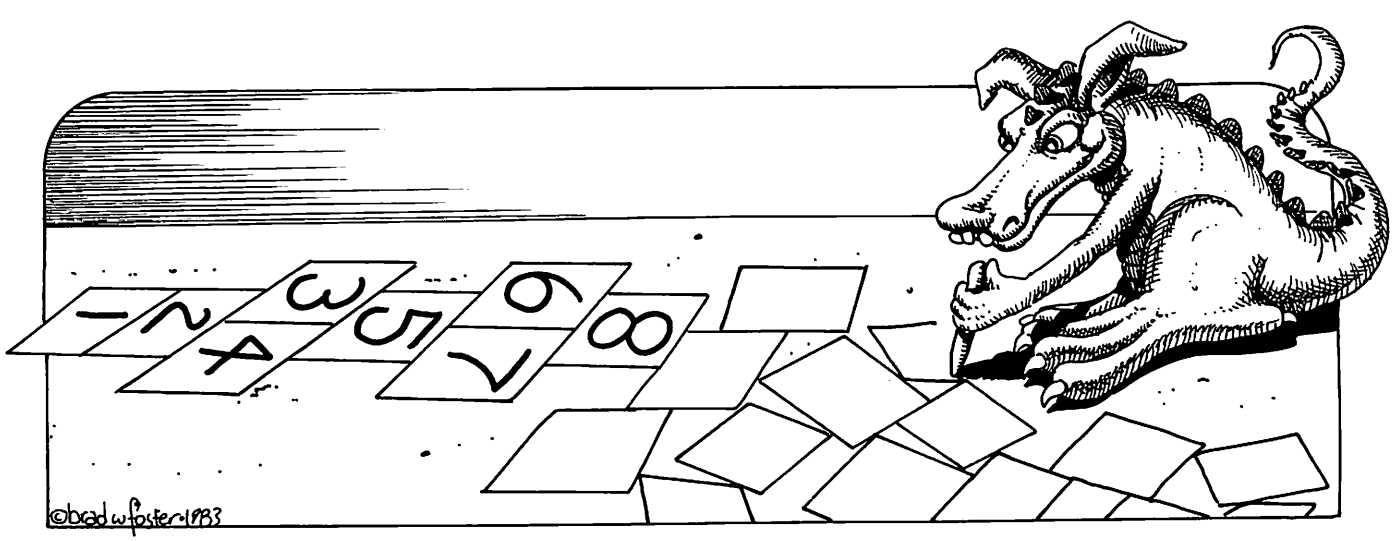


Can you make a quilt out of SQ2?

```
TO SQ2
SQ 20
RT 30
TRI 20
TRI 15
TRI 5
LT 30
END
```



Can you design some quilts of your own? Start with a square and put a design in it. Then tessellate the square.



Patterns with Tiles

TI has a graphics tool that no other LOGO has: tiles. If you have never experimented with tiles, you should! In the appendix is a table of characters and their assigned tile numbers. You can place any tile anywhere on the screen by using the command PUTTILE or PT followed by the tile number, the column number and the row number. The screen is divided into 32 columns, number 0 through 31, and 24 rows, numbered 0 through 23. If you give the command

```
PUTTILE 68 14 8
```

the letter D will appear in the 14th column of row 8. (Try writing your name in the center of the screen with tiles.)

Let's use tiles to design some repeating patterns that cover the entire screen. The easiest way is to use TILE 32. Type NOTURTLE and then MAKECHAR 32. When you press enter, you should see an 8 x 8 grid in the upper left-hand corner. To move the cursor, use the arrow keys. To fill in spaces, hold the Shift key (TI 99/4) or FCTN key (TI 99/4A) down. As you fill in spaces, you'll be able to watch your design being repeated many times on the screen. 768 times, to be exact! (32 x 24)

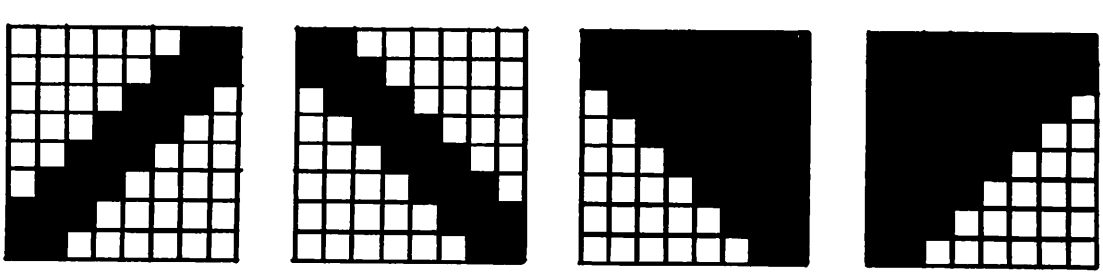
Try several designs and when you find one you like, define it by typing Shift Z (TI 99/4) or FCTN 9 (TI 99/4A).

How about adding a little color? Try TELL TILE 32 SC [5 15]
The first number is the foreground color and the second is the background color. We could even write a procedure to have the pattern blink.

```
TO BLINK
TELL TILE 32 SC [6 15]
WAIT 20 SC [15 6] WAIT 20
BLINK
END
```

To return to a plain screen, you'll have to redefine TILE 32. Type MAKECHAR 32 and "erase" your design. Then type SC [7 7].

Suppose we wanted to make a larger pattern and have it repeat? We could put several tiles together. Use graph pattern to plan your design. Here is a plan for a basket made of four tiles put together.



Type MAKECHAR 100 and draw the left side of the handle. Use MAKECHAR 101 for the right side of the handle. MAKECHAR 102 and 103 form the basket.

Now our problem is to write a procedure to put the four tiles together and have the basket repeat all over the screen. It might be easier to break it down into HANDLES and BASKETS. The left side and right side should be repeated on alternate columns and then the handles and baskets should repeat on alternate rows.

Look at the following procedure:

```
TO HANDLES  :COL  :ROW
IF  :COL > 31 MAKE "COL  :COL - 31 MAKE "ROW  :ROW + 2
IF  :ROW > 23 STOP
PUTTILE 100  :COL  :ROW
PUTTILE 101  :COL + 1  :ROW
HANDLES  :COL + 2  :ROW
END
```

If we give the command HANDLES 0 0, tile 100 will appear in column 0 row 0. The next command is PUTTILE 101 :COL + 1 :ROW, so tile 101 will appear in column 1 row 0. The recursive line adds 2 to the value of :COL, so tile 100 will be placed in column 3 row 0. When the value of :COL is greater than 31, the procedure will subtract 31 from the value of :COL and will add 2 to the value of :ROW. Thus the next row of handles will be placed on row 2. When the value of :ROW is greater than 23, the procedure will stop.

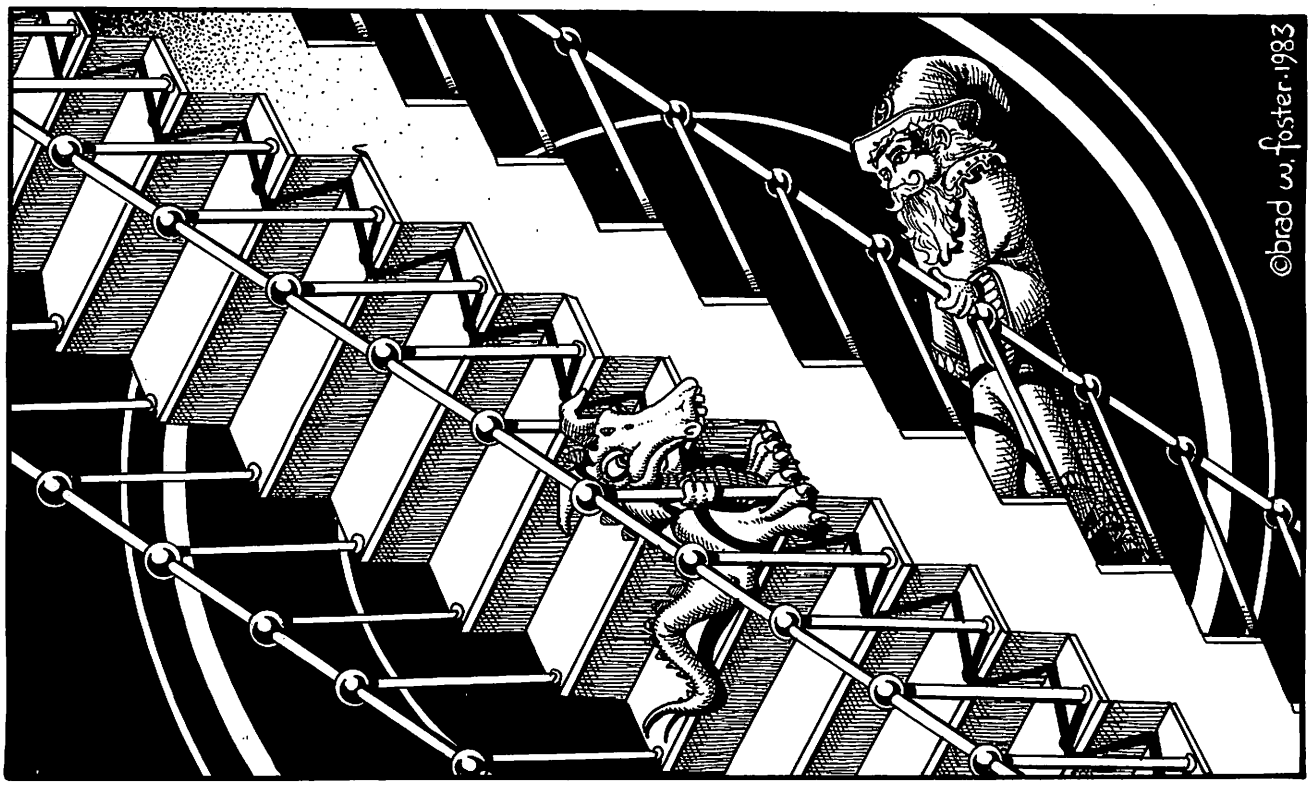
BASKETS can be similarly defined:

```
TO BASKETS  :COL  :ROW
IF  :COL > 31 MAKE "COL  :COL - 31 MAKE "ROW  :ROW + 2
IF  :ROW > 23 STOP
PT 102  :COL  :ROW
PT 103  :COL  :ROW
BASKETS  :COL + 2  :ROW
END
```

And we'll define one last procedure to put it all together and add some color:

```
TO BASKET.WEAVING
TO TELL TILE 100
SC [3 13]
HANDLES 0 0
BASKETS 0 1
END
```

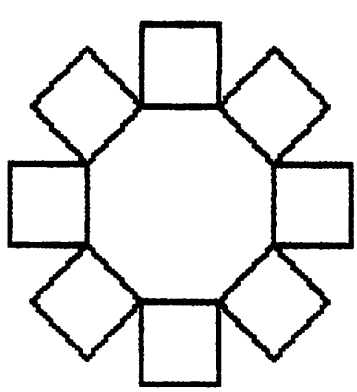
Try designing another repeating pattern with tiles.



Another kind of pattern that's fun to play around with is the optical illusion, or changing pattern. You look at it one minute and certain shapes seem to stand out. And then it seems to shift before your very eyes. Be on the lookout for optical illusions you'd like to try on the computer. (A word of warning — they can drive you slowly insane if you don't have a lot of patience or if you have a serious case of astigmastism.)

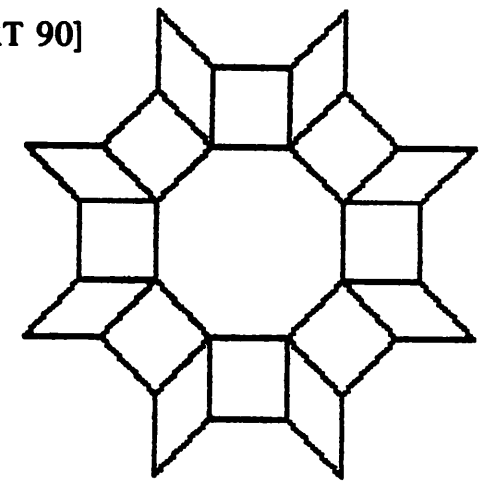
This pattern starts out fairly simply and you can quit anytime you think you're done. It starts with a ring of squares around an octagon. (Sorry, you can't quit yet.)

```
TO START
SXY 0 20
SH 0
END

TO SQ.OCT
REPEAT 8 [SQ 15 RT 90 FD 15 RT 45 LT 90]
END

TO SQ  :N
REPEAT 4 [FD  :N RT 90]
END
```

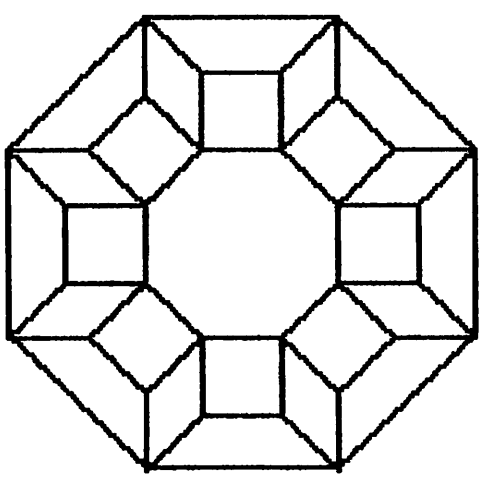


Then add a ring of rhombuses. (A rhombus is a parallelogram with four equal sides.)

```
TO RHOMBUS.RING
REPEAT 8 [RHOMBUS 15 LT 135 FD 15 RT 90]
END

TO RHOMBUS  :N
REPEAT 2 [FD  :N LT 45 FD  :N LT 135]
END
```



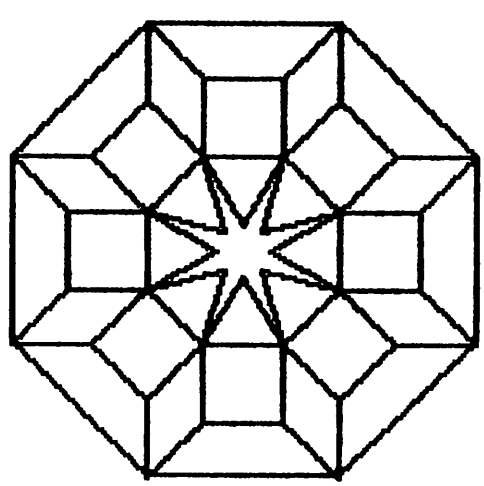
And then add an octagon around the outside. Now it has become a three-dimensional optical illusion.



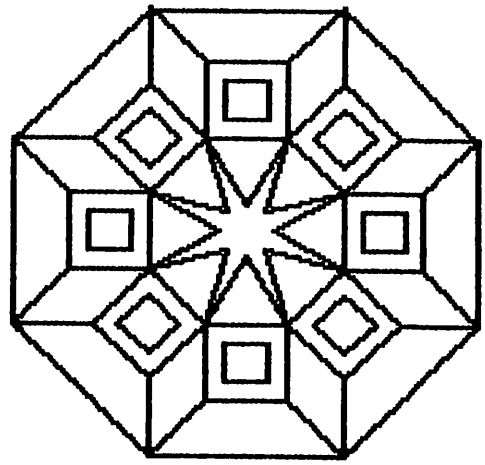
```
TO OCTAGON
FD 15 LT 45 FD 15 RT 135
REPEAT 8 [FD 36 RT 45]
END
```

And now for some real fun! Add some triangles inside to form an eight-pointed star:

```
TO STAR
START
RT 90
REPEAT 8 [TRI 15 FD 15 RT 45]
END

TO TRI  :N
REPEAT 3 [FD  :N RT 120]
END
```

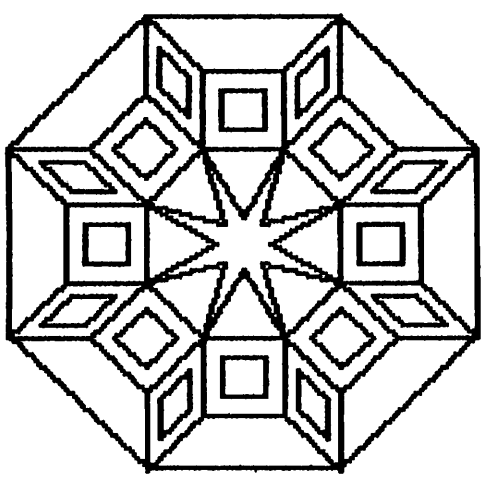


And some small squares inside the big ones:



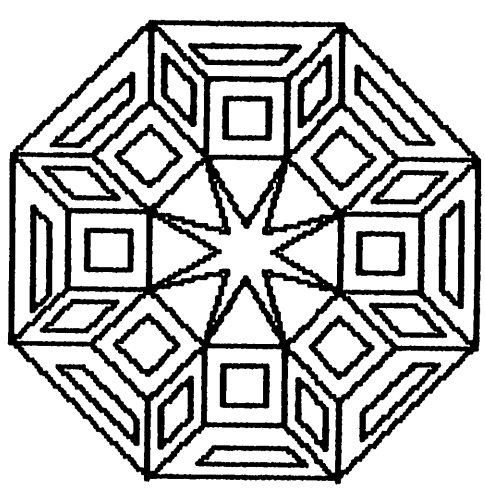
```
TO SMALL.SQUARES
START
PU FD 4 RT 90 FD 4 LT 90 PD
REPEAT 8 [SQ 7 MOVE1]
END

TO MOVE1
PU BK 4 RT 90 FD 11 LT 45
FD 4 RT 90 FD 4 LT 90 PD
END
```

And some shall rhombuses inside the big rhombuses:



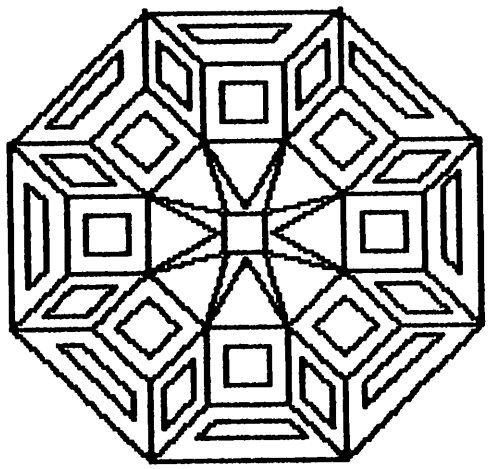
```
TO SMALL.RHOMBUSES
START
PU FD 4 LT 45 FD 4 RT 45 PD
REPEAT 8 [RHOMBUS 7 MOVE2]
END

TO MOVE2
PU BK 4 LT 45 BK 4 LT 90
FD 15 RT 45 FD 4 RT 45 FD 4 PD
END
```

And some small trapezoids inside the big trapezoids:



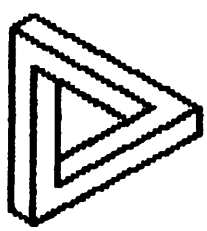
```
TO SMALL.TRAPZ
START
PU FD 19 LT 45 PD
REPEAT 8 [TRAPEZOID MOVE3]
END

TO TRAPEZOID
FD 4 RT 135 FD 20 RT 135
FD 4 RT 45 FD 14
END

TO MOVE3
PU RT 90 BK 19 RT 90 FD 15
LT 45 FD 19 LT 45 PD
END
```

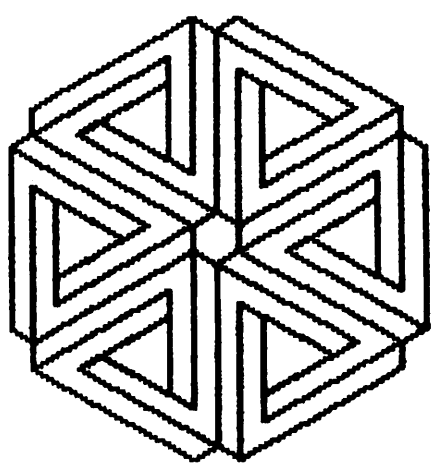
And finally, a little square in the middle:

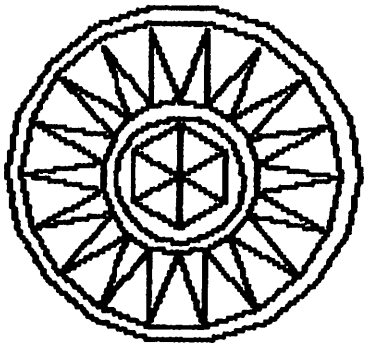


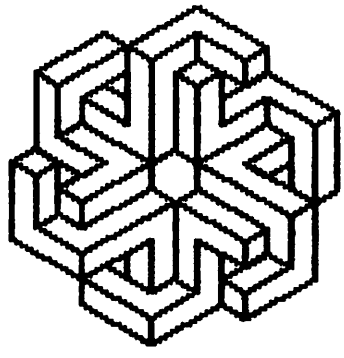
```
TO LITTLE.SQ
START
RT 165 FD 15 LT 75 SQ 8
END

TO DESIGN
START HT
SQ.OCT
RHOMBUS.RING
OCTAGON
STAR
SMALL.SQUARES
SMALL.RHOMBUSES
SMALL.TRAPZ
LITTLE.SQ
END
```

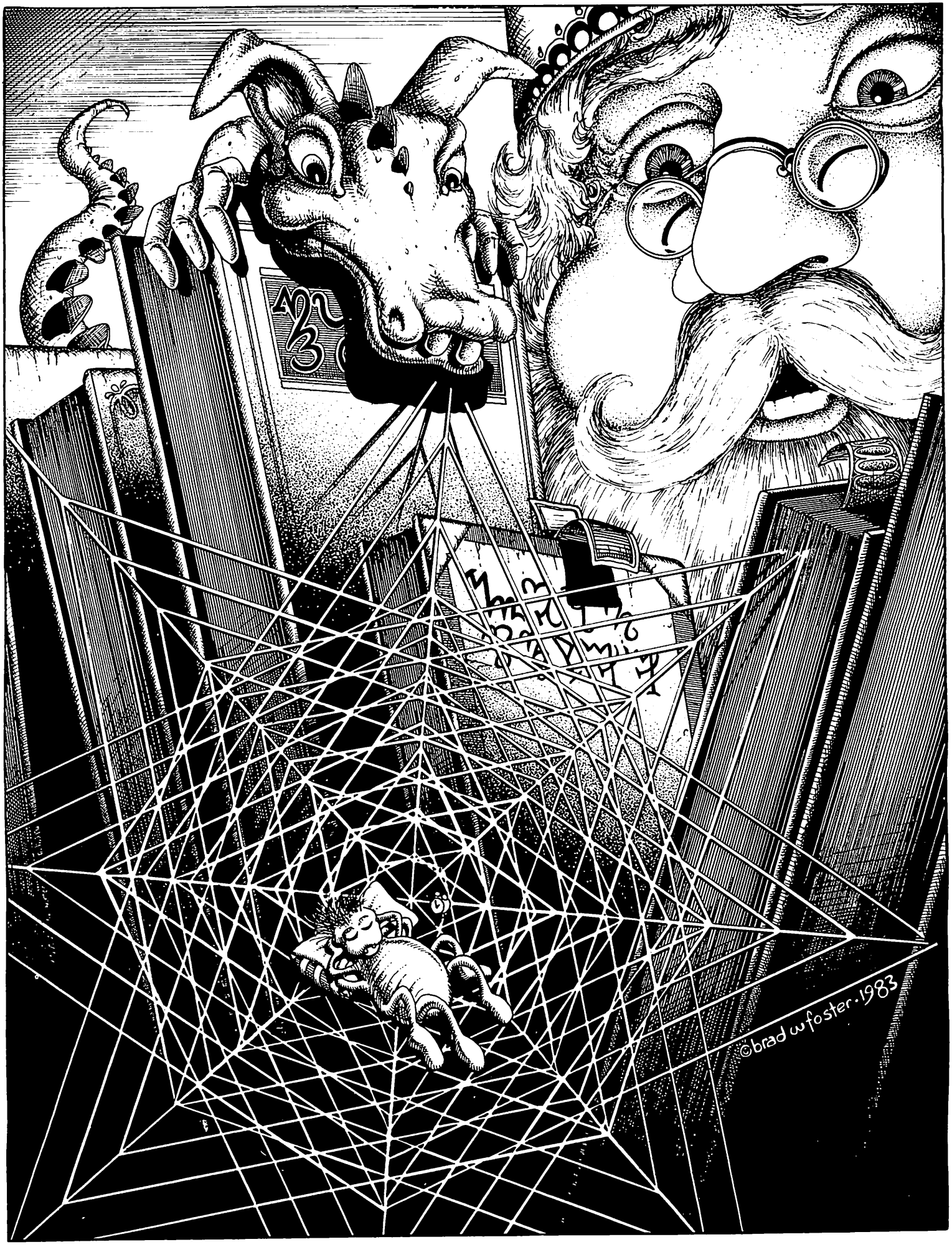
Whew! As for the next one — you're on your own, Kid! Here are a few ideas. (If you get really stumped, take a peek in the appendix for a few solutions.)

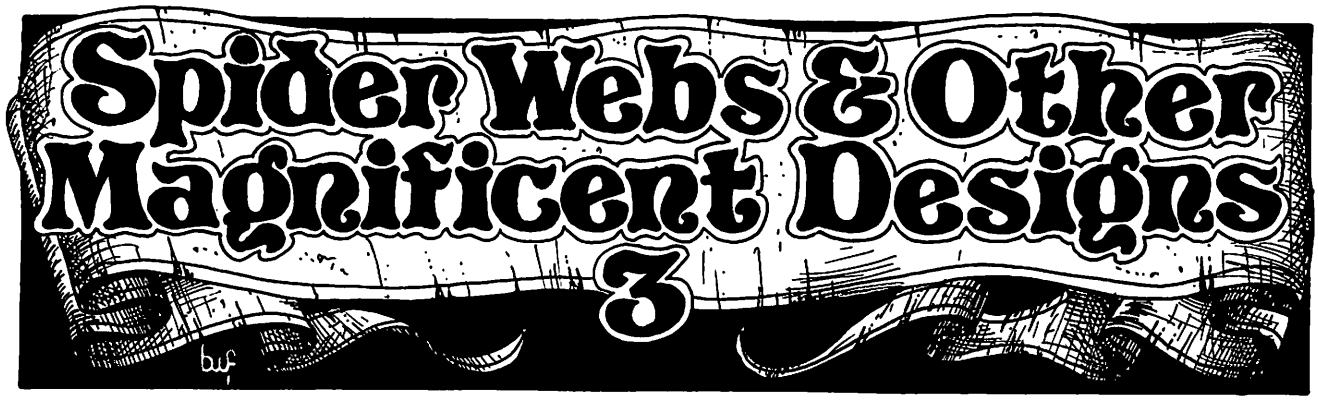









©brad w foster. 1983



"Wow!" said Little Bit, his eyes wide with wonder. "How did he do that?"

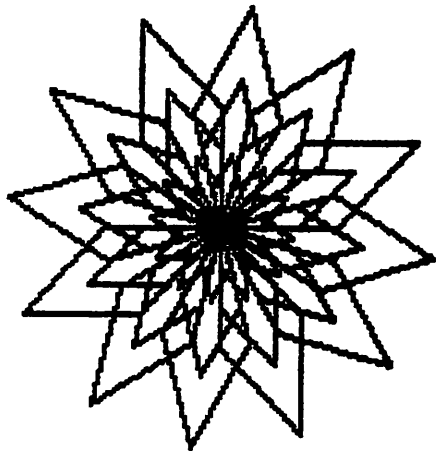
"That" was a magnificient spider web Aristotle discovered as he was searching for a book early one morning.

"I'm not sure," Aristotle replied, "but it looks like a lot of interesting angles and lines. Let's see if we can figure it out."

Except for a short break for a lunch of watercress and radish sandwiches and homemade soup, they worked the rest of the day. By mid-afternoon, they had written procedures for the spider web and had explored other lines and designs that seem to curve.

First make a rhombus:

```
TO RHOMBUS  :N
REPEAT 2 [FD  :N LT 45 FD  :N LT 135]
END
```



Then put several together to make POINTS:

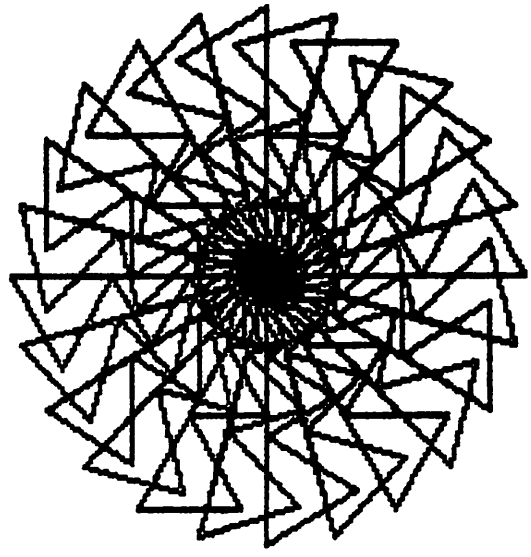
```
TO POINTS
RHOMBUS 10
RHOMBUS 20
RHOMBUS 30
END
```

Then rotate the POINTS to make the WEB:

```
TO WEB
REPEAT 12 [POINTS RT 30]
END
```

Here's another WEB made from triangles.

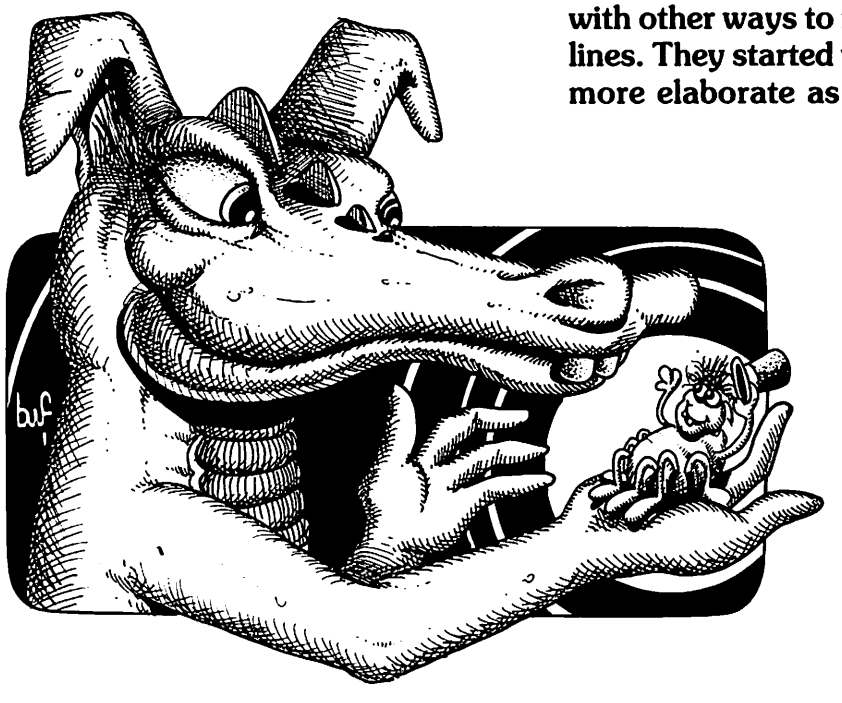
```
TO TRI  :N
REPEAT 3 [FD  :N RT 120]
END

TO TRIS  :N
IF  :N > 20 STOP
TRI  :N
RT 60 FD  :N LT 60
TRIS  :N + 5
END

TO SPOKE
TRIS 5
RT 60 BK 50
END

TO WEB
REPEAT 24 [SPOKE RT 15]
END
```



"Look at the second spider web," said Little Bit. "We only used straight lines to draw the web, so where did the circles come from? Are you doing some more magic?"

"No," Aristotle chuckled. "As you rotate and repeat patterns, the lines seem to curve."

Try rotating some of your own designs and see if you can create some "magical circles."



Aristotle and Little Bit began to experiment with other ways to make curves out of straight lines. They started with a simple spiral and got more elaborate as the day wore on.

To make a spiral, you'll have to use recursion. A recursive procedure calls — or uses — itself over and over until you stop it with shift Z (TI 99/4) or FCTN 9 (TI 99/4A) or it runs out of ink. Or you can include a conditional command within a recursive procedure so it will stop automatically. Try some spirals without conditionals and then redefine them with conditional commands.

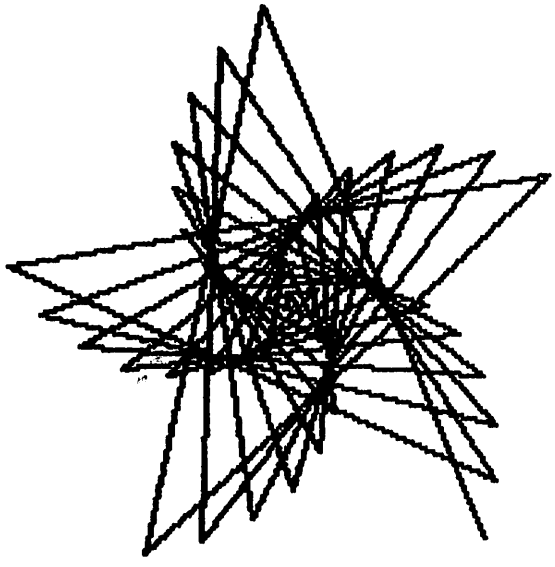
```
TO SPIRAL :X
FD :X
RT 90
SPIRAL :X + 5
END
```

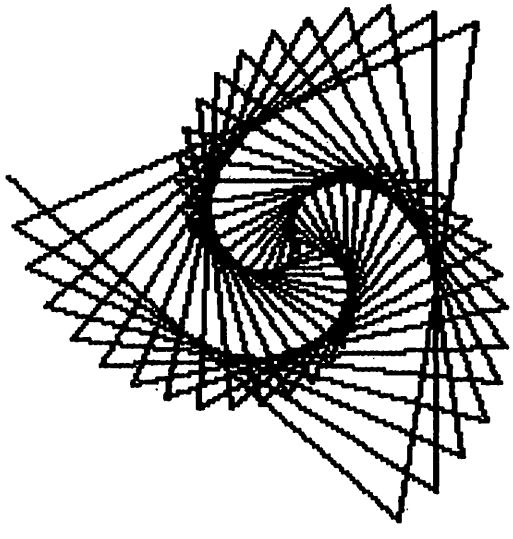
```
TO SPIRAL1 :X :A
FD :X
RT :A
SPIRAL1 :X + 5 :A + 3
END
```

```
TO SPIRAL2 :X
IF :X > 100 STOP
FD :X
RT 90
SPIRAL2 :X + 5
END
```

```
TO SPIRAL3 :X :A
IF :A > 180 STOP
FD :X
RT :A
SPIRAL3 :X + 5 :A + 3
END
```

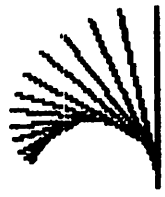
Experiment with spirals. There are an infinite number of possibilities. Try different angles. Can you make a star spiral? How about a triangular one? Try subtracting in the recursive line and starting with a large number as the input. Write down your procedures and draw what happens.





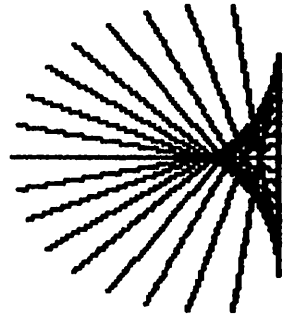
There's another way to make curves from straight lines. Think about what would happen if you made the Turtle go forward and back, then forward just a little, turn slightly, and repeat the whole sequence over and over. Try this:

```
TO CURVE :N :A
IF :N < 5 STOP
FD :N BK :N FD 5 LT :A
CURVE :N - 5 :A
END
```



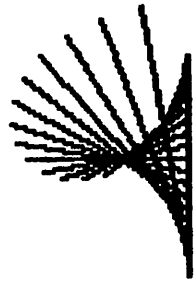
Try several inputs for :N and :A. Use large numbers for :N and small ones for :A. Then try changing the procedure slightly and see what happens. Here are a few possibilities:

```
TO CURVE1 :N :A
IF :N < 5 STOP
FD :N BK :N FD 5 LT :A
CURVE1 :N - 5 :A + 1
END
```



```
TO CURVE2 :N :A
IF :A > 180 STOP
FD 5
LT :A FD :N BK :N RT :A
CURVE2 :N :A + 10
END
```

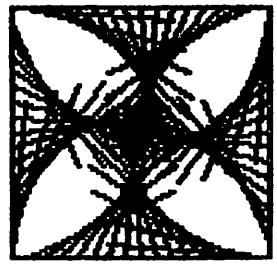
```
TO CURVE3 :N :A
IF :A > 180 STOP
FD 5
LT :A FD :N BK :N RT :A
CURVE3 :N - 5 :A +10
END
```



```
TO CURVE4 :N :A
IF :A > 180 STOP
FD 4
LT :A FD :N BK :N RT :A
CURVE4 :N :A + 12
END
```

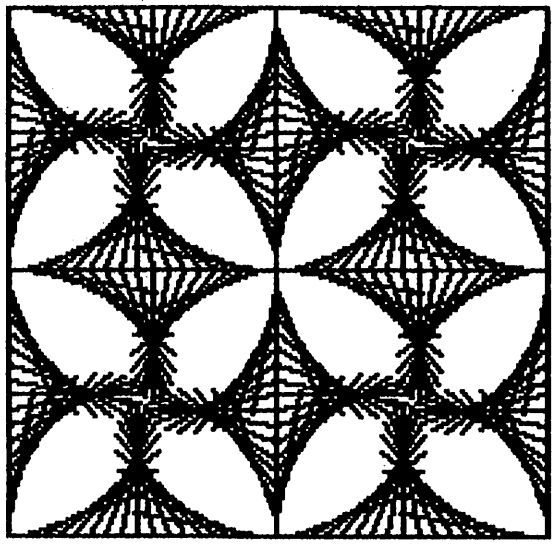
Try rotating and repeating some of the patterns. Experiment with lots of numbers to find patterns that are pleasing.

```
TO SQUARE :N :A
REPEAT 4 [CURVE4 :N :A LT 90]
END
```



This design reminded Aristotle of some of his great-grandmother's needlework, so he called it CROCHET:

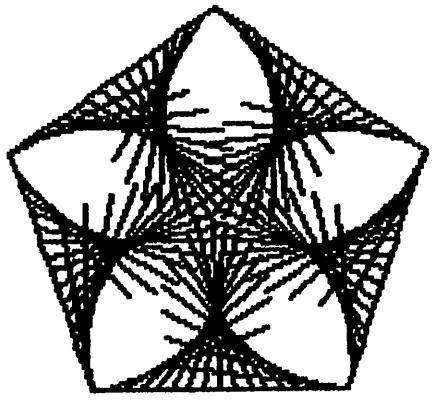
```
TO CROCHET  :N  :A
REPEAT 4 [SQUARE  :N  :A RT 90]
END
```



You could take the same original procedure, and rotate and repeat it five times instead of four to make a pentagon (with a surprise in the center). Do you remember how much to turn the Turtle to make a pentagon? (360 divided by 5 = _____)

```
TO PENTAGON  :N  :A
REPEAT 5 [CURVE4  :N  :A LT 72]
END
```

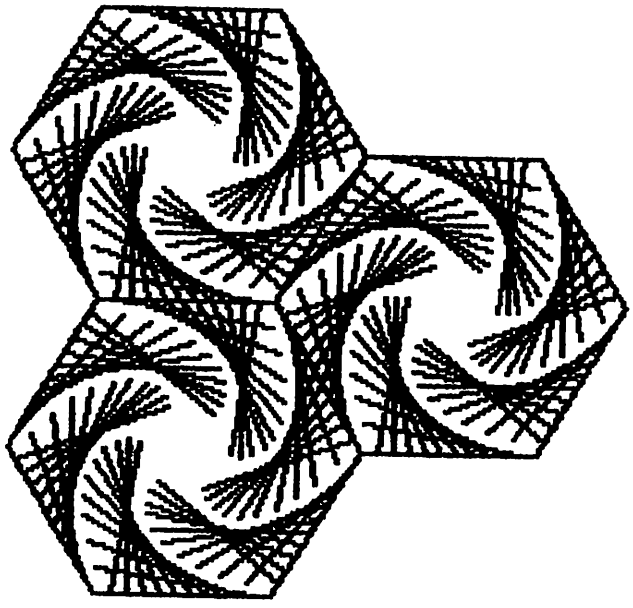
Try lots of different inputs to see what happens. This one is PENTAGON 38 8:



How would you make a hexagon?

```
TO HEXAGON

_____

END
```

Can you figure out how to rotate and repeat the hexagon to make this design?

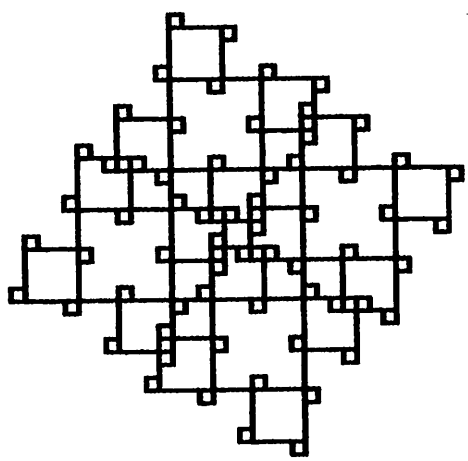


What ideas do you have? Can you make a picture frame? Can you make one design and surround it with another pattern? Experiment!

Let's take another look at recursion. Some very interesting things happen depending on where you put the recursive command. In the SPIRAL and CURVE procedures, the recursive line was the last command in the procedure. Suppose we put the recursive line within a REPEAT command? I wonder what will happen? Let's find out!

```
TO WONDER :X
IF :X < 5 STOP
REPEAT 4 [FD :X WONDER :X - 10 RT 90]
END
```

Try several different inputs and you'll create some interesting designs made of squares.

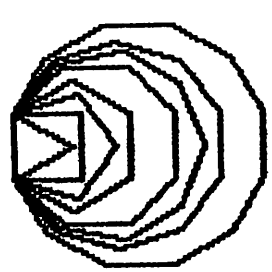


Squares are a good place to start, but don't stop there. What about triangles and hexagons and pentagons and circles? We could redefine WONDER each time we want to experiment with a different polygon. OR there might be an easier way to do it!

By now you've probably experimented with LOGO enough to know how important the number 360 is. Everytime the Turtle draws a polygon and returns to its original position, it turns through 360 degrees. To draw a square it turns 90 degrees 4 times. For a triangle, it turns 120 degrees 3 times. For a pentagon, it turns 72 degrees how many times? And so on and so on... That's called the Total Turtle Trip or the Rule of 360. We can use the T.T.T. to define a procedure to draw any polygon we want:

```
TO POLYGON :R :X
REPEAT :R [FD :X RT 360 / :R]
END
```

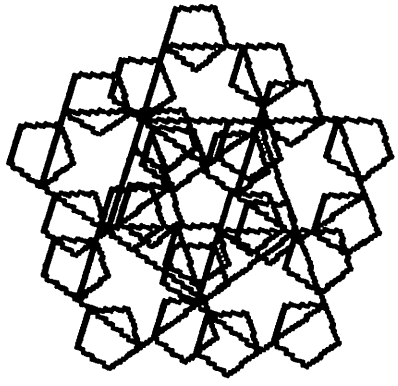
The computer does the math for us and turns RIGHT 360 divided by the number of sides of the polygon. Using the POLYGON procedure, can you draw this design?



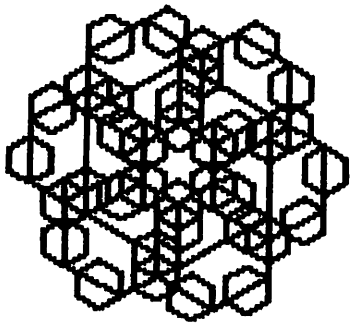
Using the Rule of 360, let's define a new WONDER procedure so we can experiment with designs made of different polygons.

```
TO WONDER1  :R  :X
IF  :X  <  5 STOP
REPEAT  :R [FD  :X WONDER1  :R  :X  -  10 RT 360  /  :R]
END
```

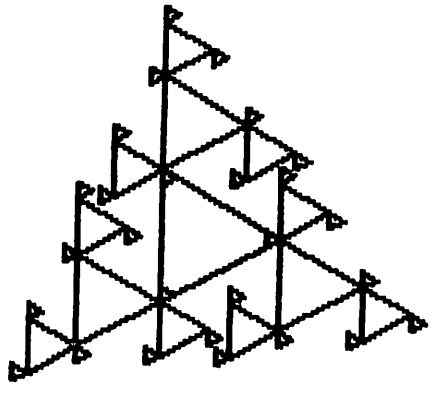
Here are a few designs. Try different inputs and write down the ones you like the best.



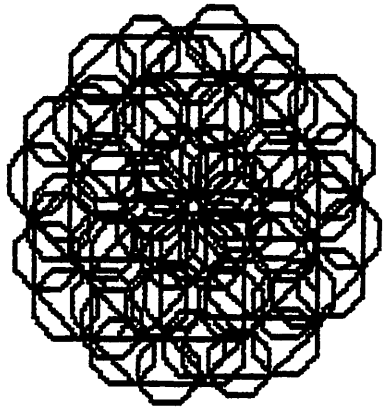
WONDER1 5 32



WONDER1 6 26



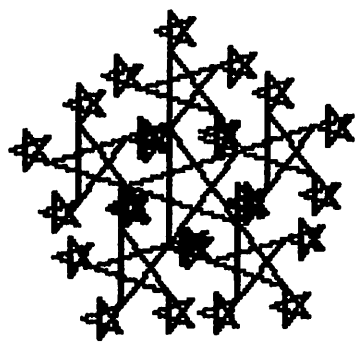
WONDER1 3 35



WONDER1 8 26

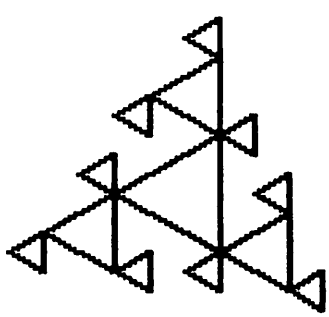
There are lots of different ways you can change the WONDER procedure. Instead of using 360 / :R, how about using 720 / :R? Instead of using :X - 10, how about using :X - 15 or :X / 2?

```
TO WONDER2 :R :X
IF :X < 5 STOP
REPEAT :R [FD :X WONDER2 :R :X - 15 RT 720 / :R]
END
```

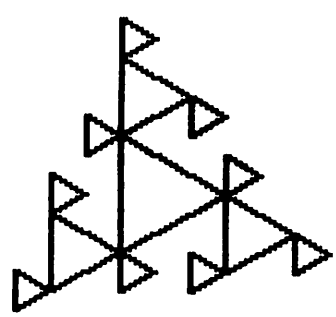


WONDER2 5 30

Can you figure out what's happening in the following two WONDER2 designs?



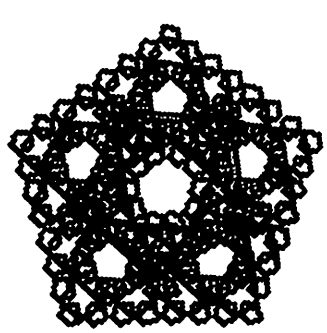
WONDER2 3 30

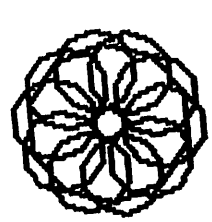


WONDER2 6 30

Try various inputs with WONDER3.

```
TO WONDER3 :R :X
IF :X < 5 STOP
REPEAT :R [FD :X WONDER3 :R :X / 2 RT 360 / :R]
END
```

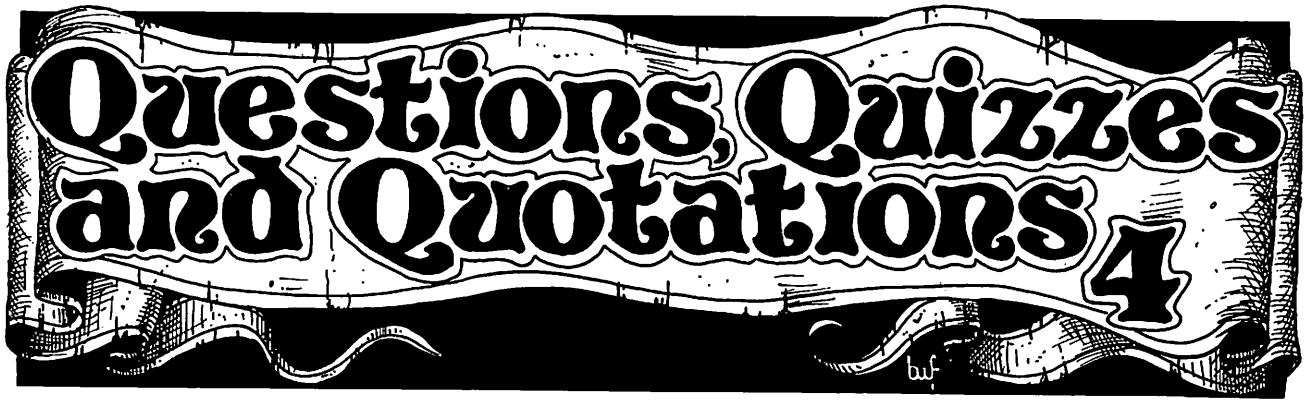




Now, I WONDER how many ways you can change the WONDER procedure...

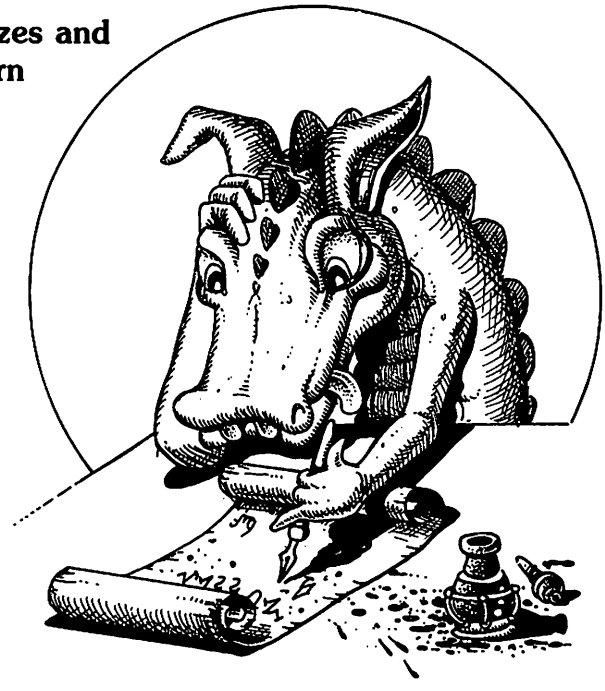
Write down your revisions and keep notes on which inputs produce the most pleasing results.

©brad w foster·1983



Aristotle loves to teach, whether it's a new magical formula he's come across or an impressive sounding word or historical facts or something about his latest hobby. He also likes to develop brain-teasers to keep Little Bit and the others on their toes.

You can learn to develop your own quizzes and brain teasers, too. First, you'll have to learn some new commands.



PRINT is a command that tells the computer to print something on the screen. Enclose what you want printed in brackets. Try a few.

```
PRINT [ARISTOTLE WAS A FAMOUS PHILOSOPHER]
PRINT [I THOUGHT HE WAS A MAGICAL WIZARD]
PRINT [ ] (Use this if you need a blank line.)
```

The PRINT command is pretty useful when you put it in a procedure. You can have the Turtle draw something on the screen and then PRINT the name of your drawing underneath it. Why don't you try it? Just add PRINT (NAME OF DRAWING] as the last command of your procedure.

SENTENCE is a command that combines words and lists. For example, if you type:

```
PRINT SENTENCE [ABRA CADABRA] [PEANUT BUTTER AND JELLY
SANDWICHES]
```

the computer will combine the two lists into one sentence.

SENTENCE can be abbreviated SE.

```
PRINT SE [ EYELASH OF GNAT] [AND WING OF BAT]
```

Sometimes you will have three or more lists or words you want combined into one sentence. In that case, enclose the command in parentheses like this:

```
PRINT (SENTENCE [PARSLEY] [SAGE] [ROSEMARY AND THYME])
```

PRINT and SENTENCE become a lot more interesting when they are used with READLINE.

READLINE enables you to begin creating interactive programs which call for some kind of response from a user. For example, the following procedure combines the user's inputs into sentences which make it seem like the computer is carrying on an intelligent conversation. Well, maybe intelligent — it all depends on what you program the computer to say.

```
TO TALK
PRINT [WHAT IS YOUR NAME?]
PRINT SENTENCE [HELLO,] READLINE
PRINT [WHO IS YOUR BEST FRIEND?]
PRINT SENTENCE READLINE [IS QUITE A CONGENIAL PERSON!]
END
```

Before we can start making up some quizzes, you'll have to know three more commands: TEST, IFT (IF True), and IFF (IF False).

TEST simply tells the computer to check an input. For example, we might have a procedure that asks someone to name the magical wizard. Then we would want to check the input to see if it was correct:

```
TEST READLINE = [ARISTOTLE]
```

There are two possibilities. The user either typed in Aristotle or he didn't. The computer will check and will then know whether to go to the IFT command or the IFF command. IF it is TRUE that the user typed ARISTOTLE, then the computer will carry out the IFT command. IF it is FALSE that the user typed ARISTOTLE, the computer will carry out the IFF command.

Look at this procedure. It's part of a FAMOUS AMERICANS quiz Aristotle is developing.

```
TO Q3
PRINT [WHO WAS THE FIRST WOMAN ASTRONAUT?]
TEST READLINE = [SALLY RIDE]
IFT PRINT [RIGHT!]
IFF PRINT [NO, IT WAS SALLY RIDE.]
END
```

Question 3 "Who was the first woman astronaut?" will appear on the screen. The computer will then wait for the user to respond to the question. If the user types "Sally Ride," the computer will respond "Right!" If the user types anything else, the computer will respond "No, it was Sally Ride."

Take a look at the other questions in the FAMOUS AMERICANS QUIZ and then add several of your own. Try them out on your friends.

Famous Americans

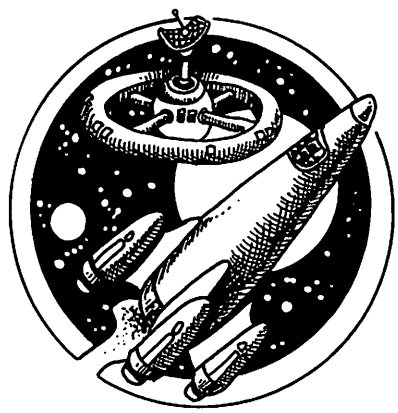
The first procedure, the superprocedure, gives the instructions on how to answer the questions and calls each of the question procedures.

```
TO FAMOUS.AMERICANS
CS
PRINT [ANSWER ALL QUESTIONS WITH]
PRINT [FIRST AND LAST NAMES]
Q1
Q2
Q3
Q4
Q5
END
```

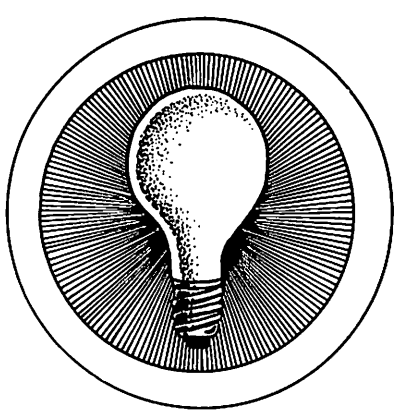
And now for the questions:

```
TO Q1
PRINT [ ]
PRINT [WHO GAVE THE GETTYSBURG ADDRESS?]
TEST READLINE = [ABRAHAM LINCOLN]
IFT PRINT [VERY GOOD!]
IFF PRINT [NO, IT WAS ABRAHAM LINCOLN.]
END
```

```
TO Q2
PRINT [ ]
PRINT [WHO WAS THE FIRST BLACK BASEBALL PLAYER IN
  THE MAJOR LEAGUE?]
TEST READLINE = [JACKIE ROBINSON]
IFT PRINT [GIVE ME FIVE!]
IFF PRINT [NOPE — IT WAS JACKIE ROBINSON.]
END

TO Q3
PRINT [ ]
PRINT [WHO WAS THE FIRST WOMAN ASTRONAUT?]
TEST READLINE = [SALLY RIDE]
IFT PRINT [RIGHT!]
IFF PRINT [NO, IT WAS SALLY RIDE.]
END
```



```
TO Q4
PRINT [ ]
PRINT [WHO INVENTED THE LIGHT BULB?]
TEST READLINE = [THOMAS EDISON]
IFT PRINT [HOW BRIGHT YOU ARE!]
IFF PRINT [IT WAS THOMAS EDISON.]
END
```



```
TO Q5
PRINT [ ]
PRINT [WHAT WAS SAMUEL CLEMENS' PEN NAME?]
TEST READLINE = [MARK TWAIN]
IFT PRINT [GOOD!]
IFF PRINT [NO, IT WAS MARK TWAIN.]
END
```

Add several questions of your own:

TO Q6

TO Q7

TO Q8

TO Q9

While Aristotle was developing the FAMOUS AMERICANS quiz, Little Bit was searching the dictionary for a vocabulary test. This is a multiple choice quiz, so the user only has to type the letter that corresponds to the correct answer. Since the answer is a single word (or, in this case, a single letter). you can use READWORD instead of READLINE. Use a single quotation mark instead of brackets, like this:

```
TEST READWORD = "A
```

READWORD is defined like this.

```
TO READWORD
OUTPUT FIRST READLINE
END
```

Vocabulary Quiz

Notice how Question5 doesn't give the user the answer. Instead, it says "TRY AGAIN" and the procedure starts over.

```
TO VOCABULARY
CS
PRINT [ANSWER THE FOLLOWING QUESTIONS]
PRINT [BY TYPING THE LETTER OF]
PRINT [THE CORRECT WORD.]
PRINT [ ]
QUESTION1
QUESTION2
QUESTION3
QUESTION4
QUESTION5
END
```

```
TO QUESTION1
PRINT [WHAT DOES GAUCHE MEAN?]
PRINT [ ]
PRINT [A. CRUDE]
PRINT [B. NICE]
PRINT [C. IN STYLE]
TEST READWORD = "A
IFT PRINT [QUITE RIGHT!]
IFF PRINT [NO, IT'S CRUDE.]
END


TO QUESTION2
PRINT [ ]
PRINT [WHAT DOES PARADOX MEAN?]
PRINT [ ]
PRINT [A. LOST SOCKS]
PRINT [B. SEEMINGLY CONTRADICTORY STATEMENT]
PRINT [C. FICTITIOUS STORY]
TEST READWORD = "B
IFT PRINT [YOU'RE RIGHT!]
IFF PRINT [NO, IT'S A STATEMENT THAT SEEMS
  CONTRADICTORY.]
END


TO QUESTION3
PRINT [ ]
PRINT [WHAT IS A TERN?]
PRINT [ ]
PRINT [A. A VERY DEEP WELL]
PRINT [B. AN EVERGREEN]
PRINT [C. A SEAGULL]
TEST READWORD = "C
IFT PRINT [YOU SURE KNOW YOUR BIRDS!]
IFF PRINT [NO, WOULD YOU BELIEVE IT'S A SEAGULL!]
END


TO QUESTION4
PRINT [ ]
PRINT [WHAT IS A SPOOF?]
PRINT [ ]
PRINT [A. A YOUNG GHOST]
PRINT [B. A HOAX]
PRINT [C. AN EATING UTENSIL]
TEST READWORD = "B
IFT PRINT [CORRECT]
IFF PRINT [NO, IT'S A HOAX.]
END
```

```
TO QUESTION5
PRINT [ ]
PRINT [WHAT DOES SNIGGLE MEAN?]
PRINT [ ]
PRINT [A. TO FISH FOR EELS]
PRINT [B. TO GIGGLE AND HUG AT THE SAME TIME]
PRINT [C. TO SNEAK COOKIES]
TEST READWORD = "A
IFT PRINT [HOW IN THE WORLD DID YOU KNOW?]
IFF PRINT [TRY AGAIN] Q5
END
```

Write some procedures of your own:

```
TO QUESTION6
```

```
TO QUESTION7
```

```
TO QUESTION8
```

Using the commands you have learned, can you develop a scrambled letters game? Have the computer list the letters of a word and see if your friends can unscramble them.

Scrambled Letters

Once the user has unscrambled the first word, we'll give him or her a choice to stop or go on after each word. First the superprocedure:

```
TO SCRAMBLE
CS
WORD1
ASK1
ASK2
ASK3
ASK4
END
```

Now scramble up some words:

```
TO WORD1
PRINT [CAN YOU UNSCRAMBLE THE]
PRINT [FOLLOWING LETTERS TO MAKE]
PRINT [A WORD?]
PRINT [ ]
PRINT [ ]
PRINT [G A I C M]
TEST READWORD = "MAGIC
IFT PRINT [RIGHT!]
IFF PRINT [THE WORD WAS "MAGIC."]
END
```

```
TO ASK1
PRINT [ ]
PRINT [CAN YOU DO ANOTHER ONE?]
TEST FIRST READWORD = "Y
IFT WORD2
IFF STOP
END

TO WORD2
PRINT [ ]
PRINT [C R O R E E S R]
TEST READWORD = "SORCERER
IFT PRINT [RIGHT]
IFF PRINT [THE WORD WAS "SORCERER."]
END

TO ASK2
PRINT [ ]
PRINT [DO YOU WANT ONE MORE?]
TEST FIRST READWORD = "Y
IFT WORD3
IFF STOP
END

TO WORD3
PRINT [ ]
PRINT [Z R W I A D]
TEST READWORD = "WIZARD
IFT PRINT [RIGHT]
IFF PRINT [THE WORD WAS "WIZARD."]
END

TO ASK3
PRINT [ ]
PRINT [HOW ABOUT ONE MORE?]
TEST FIRST READWORD = "Y
IFT WORD4
IFF STOP
END

TO WORD4
PRINT [ ]
PRINT [S L T E L T E A S O N I]
TEST READWORD = "TESSELLATION
IFT PRINT [RIGHT]
IFF PRINT [THE WORD WAS "TESSELLATION."]
END
```

```
TO ASK4
PRINT [ ]
PRINT [JUST ONE LAST ONE?]
TEST FIRST READWORD = "Y
IFT WORD5
IFF STOP
END

TO WORD5
PRINT [ ]
PRINT [C N R R E U I O S]
TEST READWORD = "RECURSION
IFT PRINT [RIGHT]
IFF PRINT [THE WORD WAS "RECURSION."]
END
```

Add some words of your own:

```
TO ASK5




TO WORD6




TO ASK6




TO WORD7
```

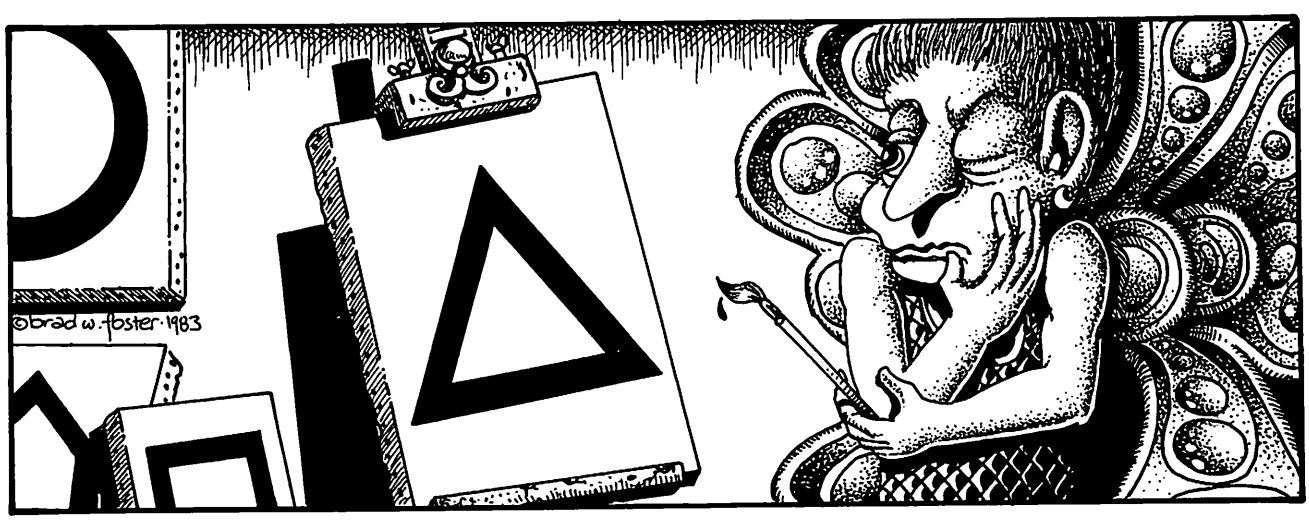
You can also combine graphics and questions in the same program. Here's one that asks someone if they want to see a particular design. If they say yes, the Turtle will draw the design. If they say no, there will be a message for them. Sometimes, you don't really want to give them a choice. In that case, you could have the computer respond with something like "Too bad. You're going to see it anyway!" and then have the Turtle go ahead and draw the design.

In this program, we're going to have the computer TEST the FIRST letter of the user's response. In response to the question "Would you like to see a tessellation?" your friend could say "yes," "yeah," "you bet," or anything starting with a Y. The command looks like this:

```
TEST FIRST READWORD = "Y
```

(We also used TEST FIRST READWORD in the SCRAMBLED LETTERS game. Did you notice it?)

This program uses a WAIT command which causes the computer to wait a specified amount of time before carrying out the next command. If we didn't include it, the designs would be erased too quickly.



Words and Designs

```
TO DESIGNS
D1
D2
D3
D4
END
```

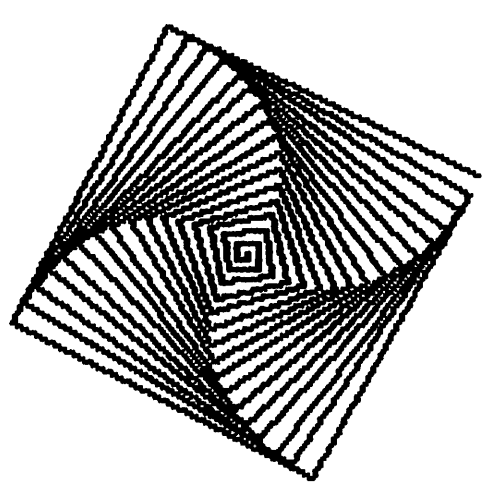
In the first procedure, we won't give them a choice whether they can see the design or not. Unless they get up and walk away, or stop it with shift Z or FCTN 9, they'll have to see the design.

```
TO D1
TELL TURTLE
HT
CS
PRINT [WOULD YOU LIKE TO SEE]
PRINT [A TESSELLATION?]
TEST FIRST READWORD = "Y
IFT TESSELLATION
IFF PRINT [AW COME ON, I WORKED REAL HARD ON IT.] WAIT 50
  TESSELLATION
END
```

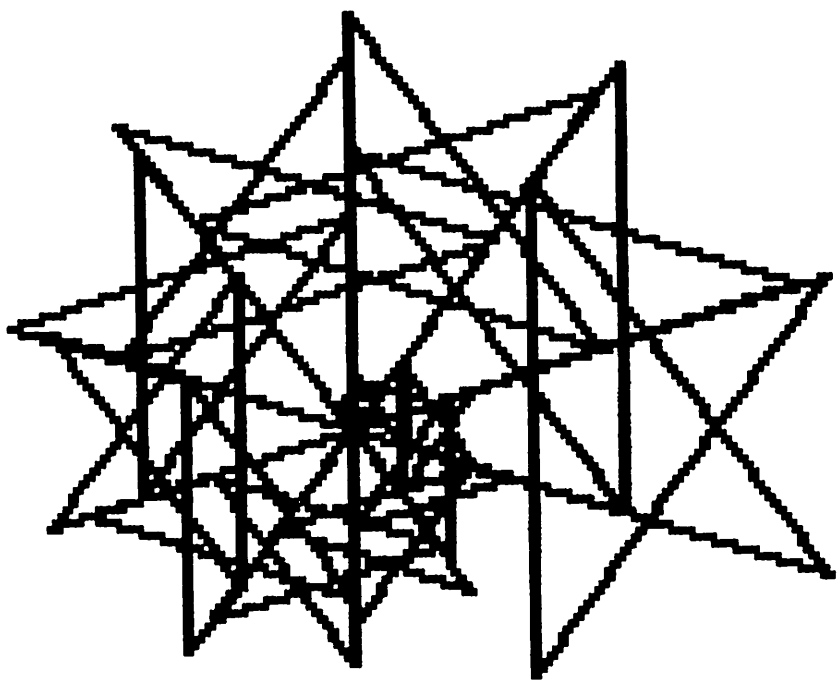
Be sure to type the entire IFF command without pressing ENTER.

For the TESSELLATION, you can use any tessellation you've designed. We used the QUILT from Chapter 2.

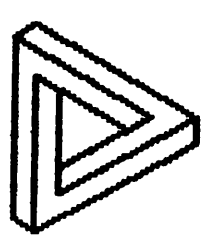
```
TO TESSELLATION
QUILT
END

TO D2
WAIT 100 CS
PRINT [HOW ABOUT A SPIRAL?]
TEST FIRST READWORD = "Y
PU FD 20 PD
IFT SPIRAL 2
IFF PRINT [OK,YOUR LOSS.]
END

TO SPIRAL  :X
IF  :X > 80 STOP
FD  :X
RT 89
SPIRAL  :X + 2
END
```



```
TO D3
WAIT 100 CS
PRINT [DO YOU LIKE STARS?]
TEST FIRST READWORD = "Y
IFT LT 90 STARS 10
IFF PRINT [TOO BAD — IT WAS A NEAT DESIGN!]
END

TO STARS  :X
IF  :X  > 80 STOP
REPEAT 5 [FD  :X RT 144] RT 36
STARS  :X + 5
END
```



```
TO D4
WAIT 100
CS
PRINT [WOULD YOU LIKE TO SEE]
PRINT [AN OPTICAL ILLUSION?]
TEST FIRST READWORD = "Y
IFT ILLUSION
IFF PRINT [GOODBYE THEN.]
END

TO ILLUSION
OP.TRI
END
```



The OP.TRI procedure is from Chapter 2. Use it or make up your own.

Add your own designs to this program! Change the responses the computer makes.

TO D5

TO D6

TO D7

©brad w foster · 1983

Somehow the nursery rhyme book had gotten knocked from the shelf and lay open on the floor.

"'Little Miss Muffet sat on a tuffet eating her curds and whey.' You've got to be kidding!" Little Bit exclaimed. "Boring, boring, boring! Let's spice it up a bit."

"Not a bad idea," said Aristotle. "What would you suggest?"

"How about 'Skinny Suzanne sat on a stepladder eating wheat bread and yogurt'?"

"Go on."

"Along came a centipede and danced and sang and frightened Suzanne to Albuquerque."

"Well, it's an improvement," said Aristotle, "but I don't think it'll be a best seller. Why don't you put it in a procedure so all those kids can make up their own stories."

So, with a Little Bit of effort, here's the Tuffet Tale. You can make it as crazy as you want.

First, you'll need to know one more command.

MAKE is a command that is a lot easier to use than to explain. You're going to MAKE something be something else. It is very helpful when you combine it with READLINE to write fractured fables. You can save inputs under certain names and then call them back whenever you want. For example, in the Tuffet Tale, we'll write a procedure that asks for the name of a girl. Then we'll use that name instead of "Little Miss Muffet." (No one's heard from her in ages anyway.)

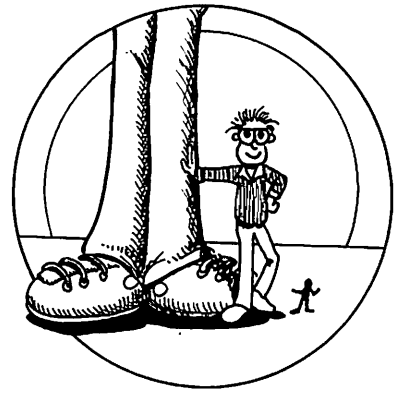
Here's how MAKE is used in the fractured fable:

```
PRINT [NAME A GIRL]
MAKE "GIRL READLINE
PRINT SENTENCE [LITTLE MISS]  :GIRL
```

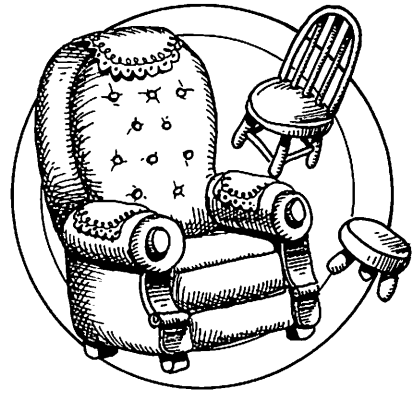
If you type in KELLI when the computer says to name a girl, the computer will store "KELLI" under the name GIRL. In the PRINT SENTENCE line, the computer will combine LITTLE MISS with the input for :GIRL and will print LITTLE MISS KELLI.

But no one likes to be called Little Miss Anything, so we'll change that, too.

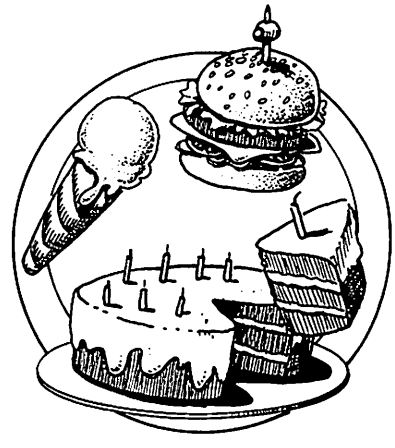
```
TO TUFFET.TALE
CS NOTURTLE
PRINT [WHAT IS YOUR NAME?]
MAKE "NAME READLINE
PRINT [NAME A SIZE FROM TINY TO GIGANTIC]
MAKE "SIZE READLINE
```



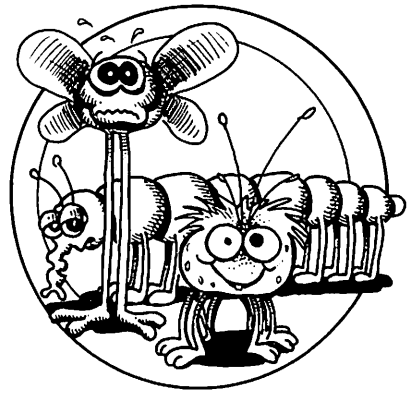
```
PRINT [NAME A GIRL]
MAKE "GIRL READLINE

PRINT [NAME SOMETHING TO SIT ON]
MAKE "SIT READLINE
```



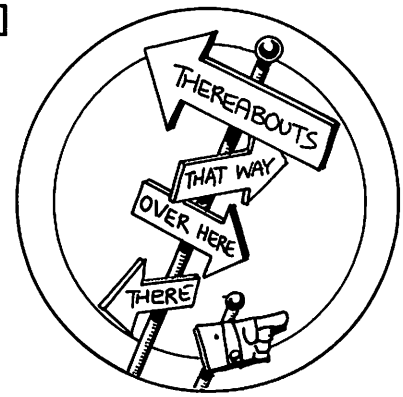
```
PRINT [NAME SOMETHING YOU LIKE TO EAT]
MAKE "YUMMYFOOD READLINE
PRINT [NAME SOMETHING YOU HATE TO EAT]
MAKE "YUCKYFOOD READLINE
```



```
PRINT [NAME A BUG, INSECT, OR OTHER SMALL CREATURE]
MAKE "BUG READLINE
```



```
PRINT [NAME TWO THINGS YOU DID YESTERDAY]
MAKE "DID READLINE

PRINT [NAME SOMEWHERE YOU'D LIKE TO GO]
MAKE "WHERE READLINE

WAIT 60
CS
PRINT [TUFFET TALE]
PRINT [ ]
PRINT SENTENCE [BY]  :NAME
PRINT [ ]
PRINT [ ]
PRINT SENTENCE  :SIZE  :GIRL
PRINT SENTENCE [SAT ON A]  :SIT
PRINT SENTENCE [EATING HER]  :YUMMYFOOD
PRINT SENTENCE [AND]  :YUMMYFOOD
PRINT SENTENCE [ALONG CAME A]  :BUG
PRINT SENTENCE [AND]  :DID
PRINT SENTENCE [AND FRIGHTENED]  :GIRL
PRINT SENTENCE [TO]  :WHERE
END
```



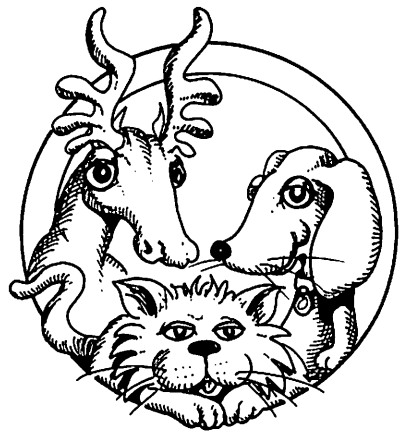
Got it? OK, go call in your friends and see who can make up the silliest Tuffet Tale. Or start from scratch and make up your own fractured fable from another nursery rhyme or poem.

Here's the beginning of 'Twas the Night before What??? See if you can complete it.

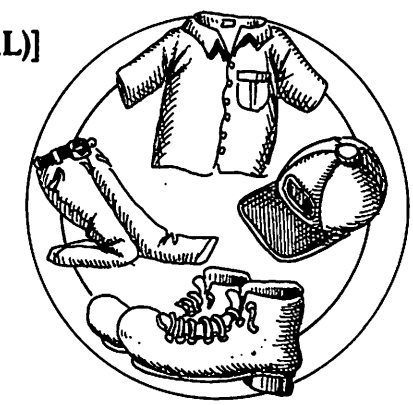
```
TO STORY
CS NOTURTLE
QUESTIONS
DRAMATIC.PAUSE
ANSWERS
END
```

```
TO QUESTIONS
PRINT [NAME A HOLIDAY]
MAKE "HOLIDAY READLINE
PRINT [NAME A BUILDING]
MAKE "BUILDING READLINE
```

```
PRINT [NAME AN ANIMAL]
MAKE "ANIMAL READLINE
```

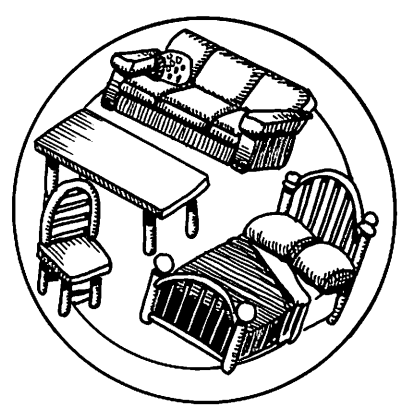


```
PRINT [NAME AN ARTICLE OF CLOTHING (PLURAL)]
MAKE "CLOTHING READLINE
```



```
PRINT [NAME A FAMOUS PERSON]
MAKE "PERSON READLINE
```

```
PRINT [NAME A PIECE OF FURNITURE (PLURAL)]
MAKE "FURNITURE READLINE
```



```
PRINT [NAME SOMETHING SWEET]
MAKE "SWEET READLINE
```

And you can go on and on...

The next few lines aren't really necessary but they add a nice dramatic pause to the program.

```
TO DRAMATIC.PAUSE
PRINT [WAIT A MINUTE WHILE I PREPARE]
PRINT [YOUR ORIGINAL COMPOSITION...]
WAIT 100
END
```

And then your semi-original poem:

```
TO ANSWERS
CS
PRINT SENTENCE [TWAS THE NIGHT BEFORE]  :HOLIDAY
PRINT SENTENCE [WHEN ALL THROUGH THE]  :BUILDING
PRINT [NOT A CREATURE WAS STIRRING]
PRINT SENTENCE [NOT EVEN A]  :ANIMAL
PRINT SENTENCE [THE]  :CLOTHING
PRINT [WERE HUNG BY THE CHIMNEY WITH CARE]
PRINT SENTENCE [IN HOPES THAT]  :PERSON
PRINT [SOON WOULD BE THERE]
PRINT [THE CHILDREN WERE NESTLED]
PRINT SENTENCE [ALL SNUG IN THEIR]  :FURNITURE
PRINT SENTENCE [WHILE VISIONS OF]  :SWEET
PRINT [DANCED IN THEIR HEADS]
END
```

Try picking out key words from other poems and making up your own procedures.

"Suppose I change some words in some famous sayings and see if you can figure them out," said Little Bit.

"Tell me more," said Aristotle.

"If I give you words that mean the same thing as the key words in the saying, you should be able to figure out the original quotation." Little Bit had that devilish look in his eye and Aristotle knew he was up to something.

"OK, try me," he said. (Aristotle was always ready for anything.)

"'A revolving rock collects no bryophytic plants'," grinned Little Bit.

"Hmm," said Aristotle as he pulled on his beard. "Revolving could mean 'spinning' or 'turning' or 'rolling' or a number of other things. Where's my Thesaurus? A rock is probably a stone. Let's see. 'A spinning stone'...no. 'A rolling stone' — that's it! 'A rolling stone gathers no moss!' Where in the world did you get 'bryophytic plants'?"

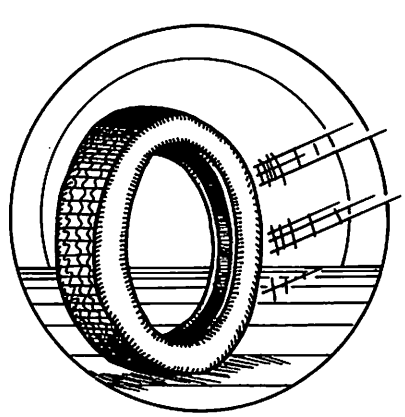
"Cincho," said Little Bit. "I looked up 'moss' in the dictionary."

"This look like it could be fun. Get your Thesaurus, your dictionary and your imagination and let's get to work!"

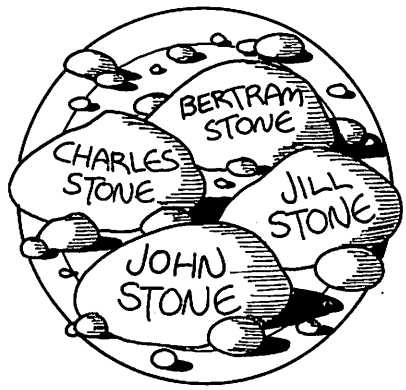
In customizing cliches, you can write procedures to give the clues and have your friends try to figure out the original saying, or you could write procedures that ask for words that mean the same thing as the key words. Then the computer would write the customized cliche using the inputs, just as it did in the fractured fables. Try it both ways. In giving clues, you could ask for a word that means the same thing or you could take a characteristic of that item and ask for something else that has the same characteristic. For example, if the word is "FIRE," you could say "Name something that is very hot."

The "rolling stone procedure" could be written like this:

```
TO CLICHE
CS NOTURTLE
PRINT [WHAT IS A SYNONYM FOR "ROLLING"?]
MAKE "ROLLING READLINE
```



```
PRINT [WHAT IS ANOTHER WORD FOR "STONE"?]
MAKE "STONE READLINE
```




```
PRINT [GIVE ME A SYNONYM FOR "GATHERS".]
MAKE "GATHERS READLINE
PRINT [WHAT IS "MOSS"?]
MAKE "MOSS READLINE
WAIT 100
CS
PRINT (SENTENCE [A]  :ROLLING  :STONE  :GATHERS [NO]  :MOSS)
END
```

Write a procedure for another cliche or two:

```
TO CLICHE2
```

```
TO CLICHE3
```

You could reverse the process — give the clues and see if your friends can figure out the cliche. Use your dictionary and make them interesting. Here's the rolling stone procedure done this way:

```
TO CLICHE4
CS
PRINT [CAN YOU FIGURE OUT THIS SAYING?]
PRINT [ ]
PRINT [A REVOLVING ROCK COLLECTS NO BRYOPHYTIC PLANTS]
TEST READLINE = [A ROLLING STONE GATHERS NO MOSS]
IFT PRINT [RIGHT!]
IFF PRINT [TRY AGAIN] WAIT 100 CLICHE4
END
```

Your turn:

```
TO CLICHE5
```

```
TO CLICHE6
```

Write procedures for several cliches and then put them together in a superprocedure:

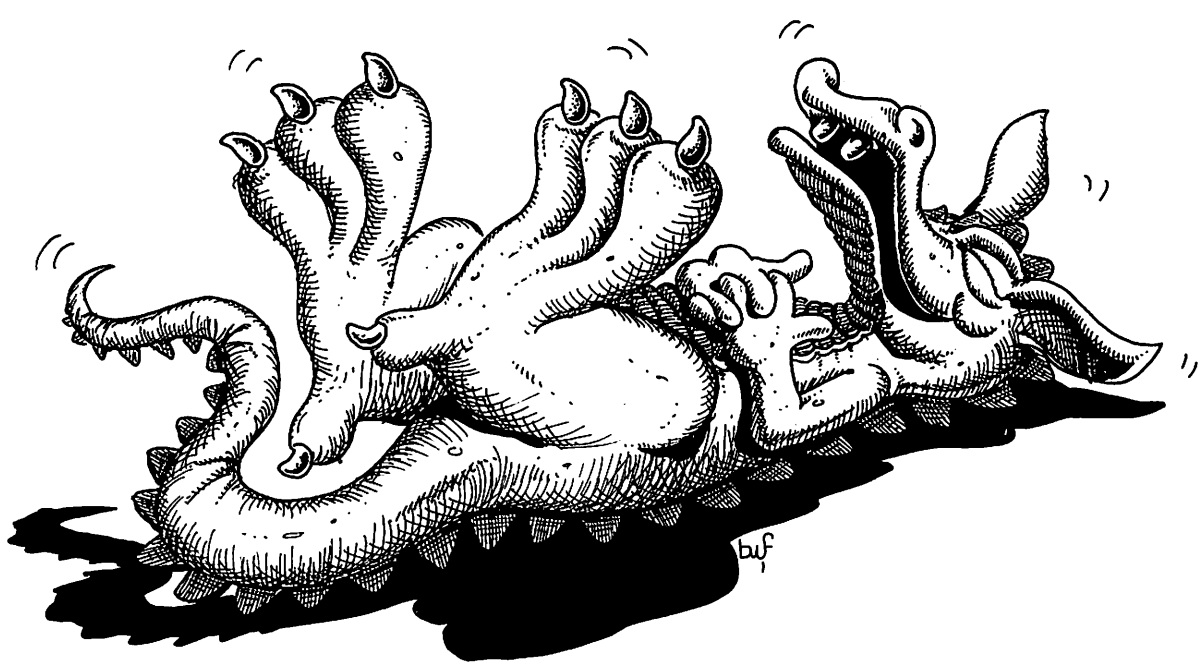
```
TO CLICHES
CS NOTURTLE
CLICHE1
WAIT 300
CLICHE2
WAIT 300
CLICHE3
WAIT 300
CLICHE4
WAIT 300
CLICHE5
WAIT 300
CLICHE6
END
```

Try this one:

```
TO CLICHE7
CS
PRINT [WHAT IS THE OPPOSITE OF "ANYTHING"?]
MAKE "ANYTHING READLINE
PRINT [WHAT IS THE ANTONYM OF "CAN"?]
MAKE "CAN READLINE
PRINT [GIVE ME AN ANTONYM FOR "WRONG".]
MAKE "WRONG READLINE
PRINT [WHAT IS THE OPPOSITE OF "WILL"?]
MAKE "WILL READLINE
WAIT 100
CS
PRINT (SENTENCE [IF]  :ANYTHING  :CAN [GO]  :WRONG)
PRINT SENTENCE [IT]  :WILL
END
```

"Boy, have I got one for you," said Little Bit. He tried to hide the volume of Shakespeare, but Aristotle glimpsed the first three letters of the title. They were M A C.

"Quadruple work and hot water; Lava smolder and a pot boil." Little Bit could hardly contain himself.



YE DARK FOREST
KEEP OUT!!
I'D GO BACK IF I WERE YOU
SPOOKY STUFF BE AHEAD!
©brad w foster 1983

"Come on, Little Bit," said Aristotle. "We're going on an adventure."

"Should I pack a bag?" asked the mischievous little dragon.

"No," said Aristotle, "just your imagination. We're going to combine a little bit of this and a little bit of that and put together an illustrated, interactive story in LOGO."

"Wow, this could get kind of complicated!"

"Well, it can get as complicated as we want it to get," Aristotle chuckled. "But I was thinking about just starting it off and then leaving it to those kids who are reading this book."

"Are you sure some of that mischief didn't spill on you, too?" Little Bit asked.

But Aristotle was already deep in thought. "One dark and dreary afternoon..."

Starting was easy enough.

```
TO START
CS NOTURTLE
TELL BG SC  :BLUE
PRINT [ONE DARK AND DREARY AFTERNOON]
PRINT [YOU SADDLE YOUR HORSE AND GO]
PRINT [FOR AN ADVENTURE IN THE FOREST.]
PRINT [ ]
PRINT [WHEN YOU GET TO THE FOREST,]
PRINT [IT LOOKS SPOOKY AND FORBIDDING.]
PRINT [ ]
PRINT [DO YOU WANT TO KEEP GOING OR TURN BACK?]
TEST READLINE = [KEEP GOING]
IFT GO.IN
IFF TURN.BACK
END
```

We've set up a choice for our adventurers to make. They can turn back or keep going. Since we don't really want them to turn back, let's define a procedure for TURN.BACK which really forces them to GO.IN.

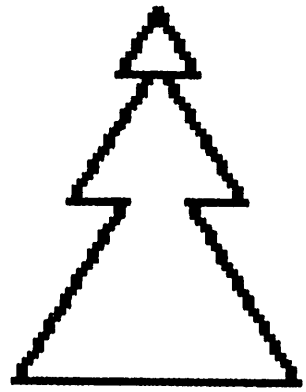
```
TO TURN.BACK
CS
PRINT [YOU LILY LIVERED COWARD!]
PRINT [ ]
PRINT [COME ON, YOU HAVE TO AT LEAST]
PRINT [ENTER THE FOREST!]
WAIT 200
GO.IN
END
```

In GO.IN, we'll draw the forest and lead the adventurer on to an old castle. Then we'll give him another choice: to explore the grounds or enter the castle. Remember the castle we drew in Chapter 1? We can add it right after the print statement "YOU COME UPON AN OLD CASTLE."

```
TO GO.IN
TELL TURTLE
HT
FOREST
WAIT 200
CS
PRINT [YOU COME UPON AN OLD CASTLE.]
CASTLE
WAIT 100
PRINT [DO YOU WANT TO GO INSIDE]
PRINT [OR EXPLORE THE GROUNDS?]
TEST READLINE = [GO INSIDE]
IFT INSIDE
IFF GROUNDS
END
```

To draw the forest, first design a tree. Little Bit's first effort was with sprites. He put a BALL on top of a ROCKET and called it a tree. You don't have to use his tree. You can design your own.

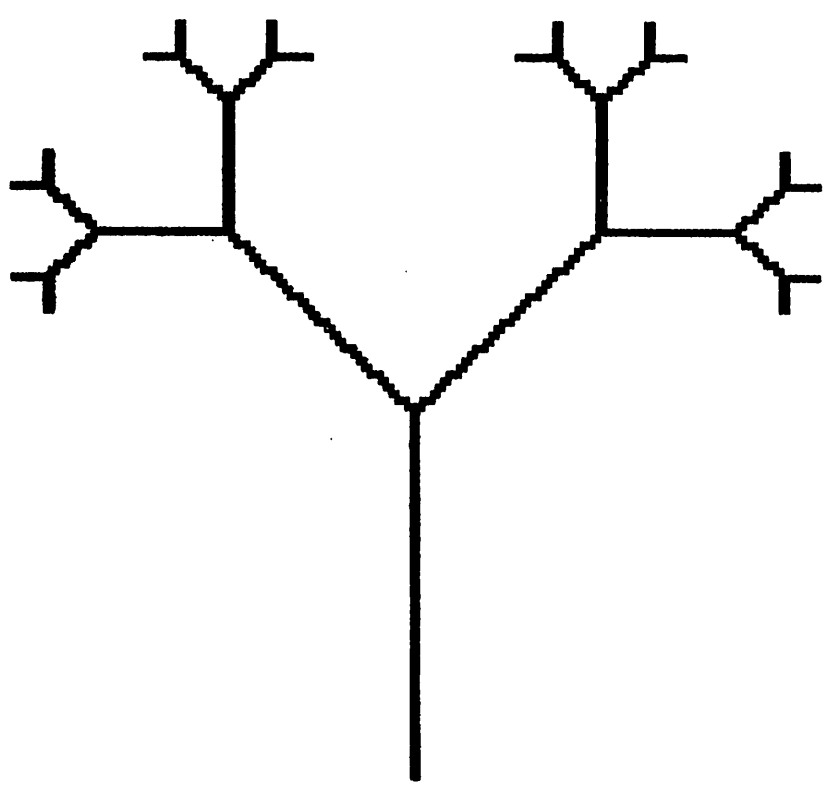
```
TO PINE
TELL TURTLE HT
SC  :GREEN
RT 30 FD 20
LT 120 FD 5
RT 120 FD 15
LT 120 FD 3
RT 120 FD 7
RT 120 FD 7
RT 120 FD 3
LT 120 FD 15
RT 120 FD 5
LT 120 FD 20
RT 120 FD 26
END
```



There are, of course, many different kinds of trees besides PINEs.

Let's look at one that has lost all its leaves. This is a classic example of LOGO recursion. Follow it one step (branch?) at a time and you'll begin to understand the power of recursion.

```
TO TREE  :X  :L
IF  :X < :L STOP
LT 45
FD  :X
TREE  :X / 2  :L
BK  :X
RT 90
FD  :X
TREE  :X / 2  :L
BK  :X
LT 45
END
```



Try TREE 50 5, and then add a trunk.

PLACE is a procedure to cause the trees to appear randomly. (Have you ever seen a natural forest with trees in nice, neat rows?) RANDOM tells the computer to select a number between 0-9. Since those numbers are rather limited, we'll simply tell the computer to multiply the random number by another number.

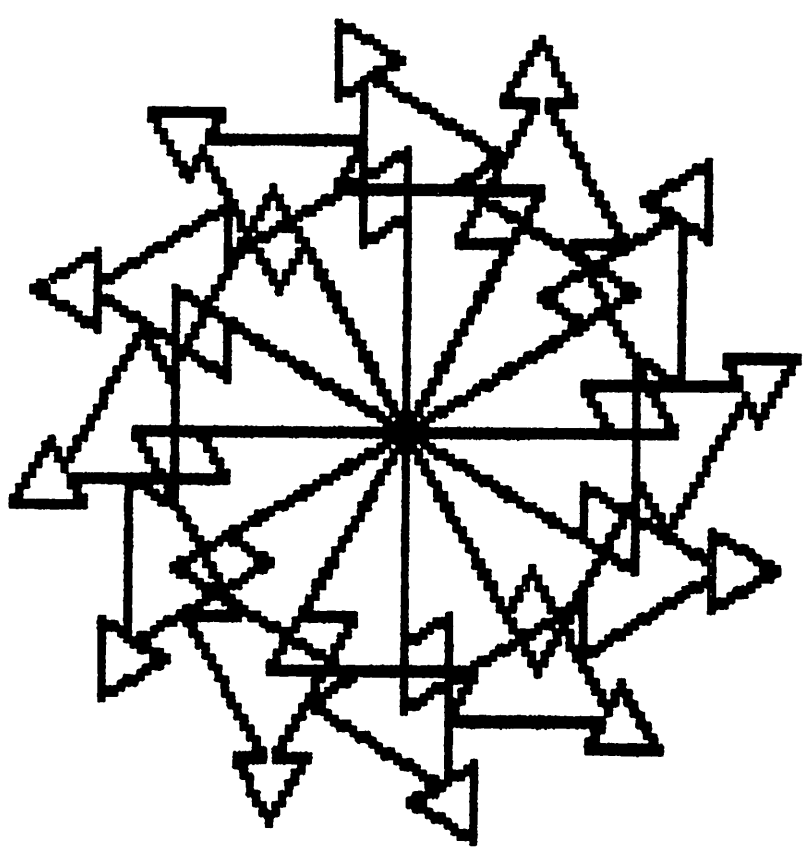
In order to select any X coordinate, we need a number between 0 and 254. If the number is greater than 127, the Turtle will wrap around to the left-hand side of the screen, so 128-254 will take care of any positions on the left. For the Y position, any number greater than 96 will cause the Turtle to wrap around to the bottom of the screen. If we use the command SXY (RANDOM * 28) (RANDOM * 22), the computer will select a number between 0 and 9 and will multiple that number by 28 for the X position. For the Y position, the random number will be multiplied by 22.

```
TO PLACE
SXY (RANDOM * 28) (RANDOM * 22)
SH 0
END
```

Now let's PLACE 10 PINEs for the FOREST:

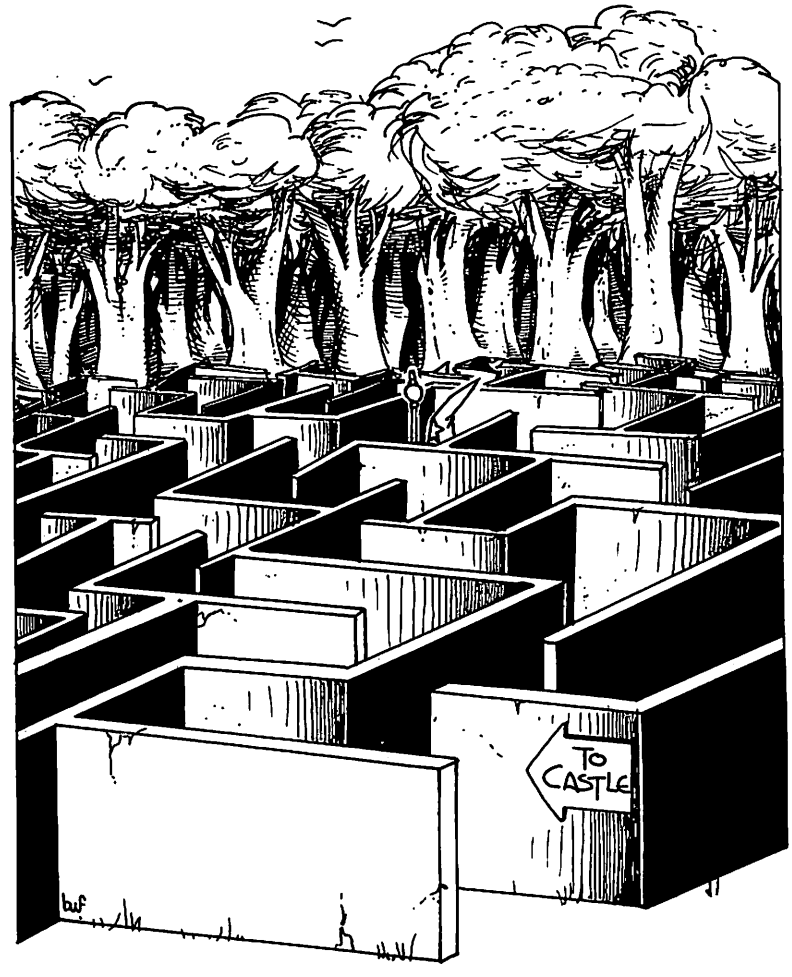
```
TO FOREST
CS
REPEAT 10 [PLACE PINE]
END
```

"Little Bit, what are you doing spinning the PINE. That's not the way to make a forest." (Can you figure out how to make this design?)



We set up another decision in GO.IN, so we'll have to define different procedures to take care of the two possible choices. One will be to explore the GROUNDS and the other will be to go into the CASTLE.

In GROUNDS, the adventurer will find himself in the middle of a maze and will have to move himself (the Turtle) out of the maze with single key-stroke commands. Then we'll give him another chance to go into the castle.



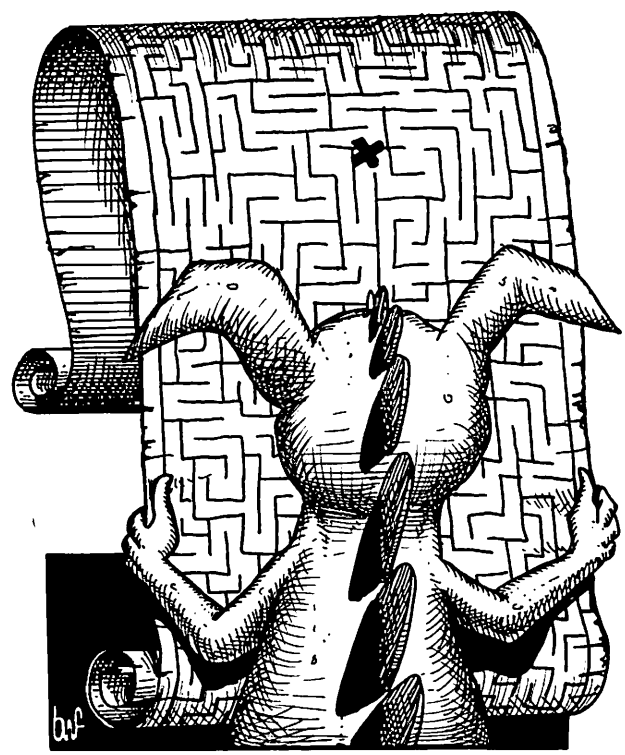

```
TO GROUNDS
CS
ST
SXY -80 (-30)
SPI 100
PRINT [YOU'VE GOTTEN LOST IN A MAZE]
WAIT 100
PRINT [TO MOVE:  USE "M". "R" = RT; "L" = LT.]
PRINT [WHEN YOU'RE OUT, TYPE "O"]
END
```

Just for fun, let's do a rectangular spiral for the maze.

```
TO SPI  :X
SC  :BLACK
IF  :X < 10 STOP
FD  :X
RT 90
FD  :X * 2
RT 90
SPI  :X - 15
END
```

Here are the single keystroke command procedures:

```
TO R
PU RT 30 PD
END

TO L
PU LT 30 PD
END

TO M
PU FD 5 PD
END
```



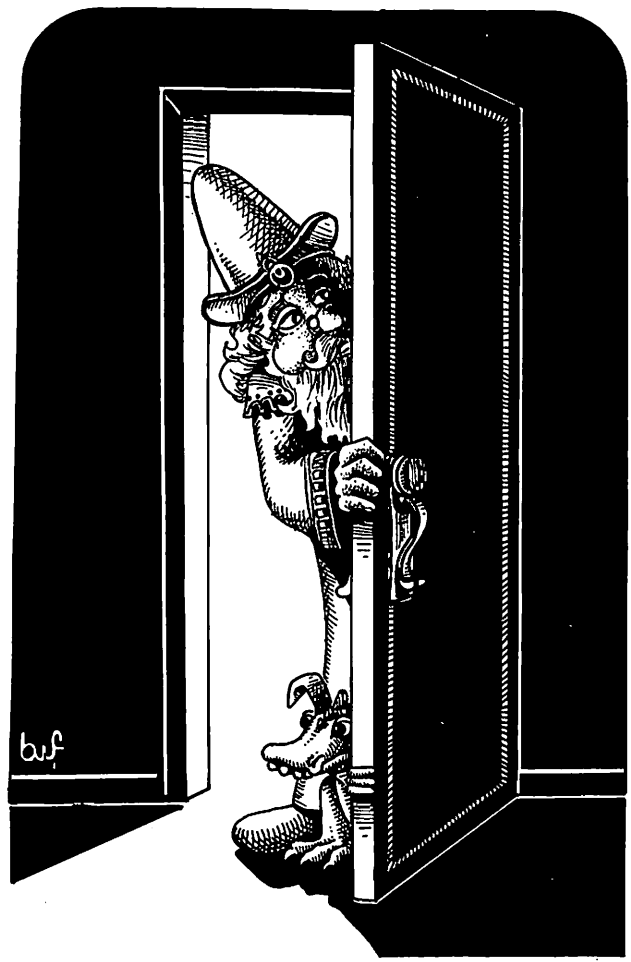
As our adventurer is making his way through the maze, the computer is in the immediate mode. We need to get the control back into a procedure. That's why we told him to type "O" when he gets out of the maze.

```
TO O
CS NOTURTLE
PRINT [WELL, I'M GLAD YOU MADE IT OUT!]
PRINT [DO YOU WANT TO GO INSIDE THE CASTLE?]
TEST FIRST READWORD = "Y
IFT INSIDE
IFF PRINT [YOU LILY LIVERED COWARD!] STOP
END
```

We defined READWORD earlier as:

```
TO READWORD
OUTPUT FIRST READLINE
END
```

IF he chooses to turn back this time, it's his loss. We'll just stop the action. Meanwhile, the rest of us will go INSIDE.



```
TO INSIDE
CS NOTURTLE
PRINT [AS YOU OPENED THE DOOR]
PRINT [IT SCREECHED ON ITS HINGES.]
PRINT [ ]
PRINT [YOU HEAR FOOTSTEPS COMING.]
PRINT [ ]
PRINT [QUICK! DO YOU WANT TO GO UPSTAIRS]
PRINT [OR DOWN TO THE CELLAR?]
TEST READLINE = [UPSTAIRS]
IFT UPSTAIRS
IFF CELLAR
END
```

Let's work on the CELLAR first. Suppose there's a big supply of fireworks in the cellar and somehow they start going off!

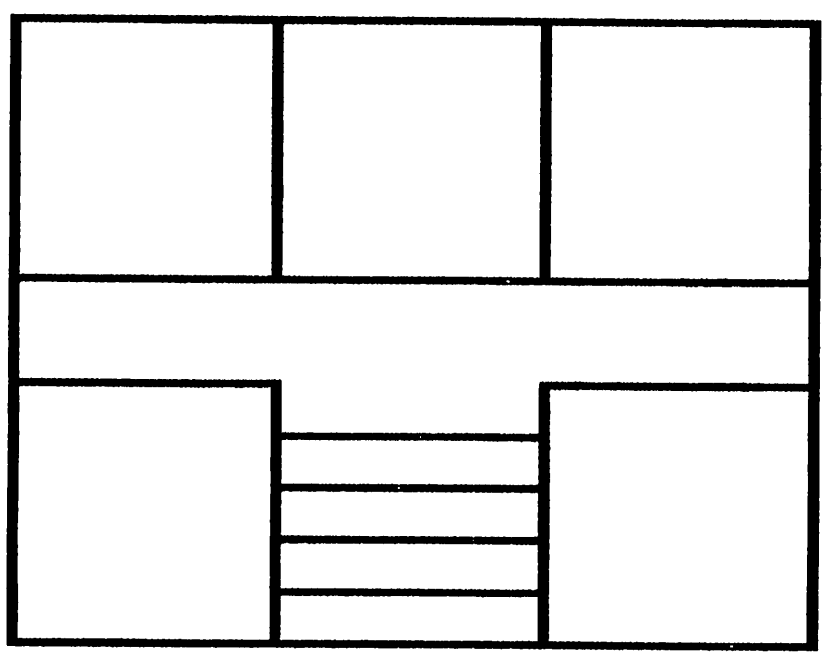
```
TO CELLAR
CS
PRINT [AS YOU OPENED THE CELLAR DOOR,]
PRINT [IT SET OFF AN ALARM]
PRINT [WHICH SET OFF SOME FIREWORKS]
WAIT 200
TELL BG SC  :BLACK
TELL TURTLE HT
FIREWORKS 5
WAIT 100 CS
TELL BG SC  :CYAN
PRINT [YOU'D BETTER RUN UPSTAIRS!]
WAIT 200
UPSTAIRS
END
```

FIREWORKS has one conditional to change the pencolor and another to check the length of the line. Can you predict what this procedure will do before you try it?

```
TO FIREWORKS  :T
IF  :T = 0 STOP
CS
SXY (RANDOM * 28) (RANDOM * 22)
STARS 1 5
FIREWORKS  :T - 1
END

TO STARS  :C  :D
IF  :C = 16 MAKE "C  :C - 16
IF  :D > 100 STOP
SC  :C
FD  :D RT 144
STARS  :C + 1  :D + 5
END
```

We have "gently persuaded" our adventurers to go UPSTAIRS. We'll draw a FLOORPLAN and let them select a room. But they've got to hurry because someone is after them.

```
TO UPSTAIRS
CS
FLOORPLAN
CHOOSE
END

TO FLOORPLAN
TELL TURTLE HT
SXY -50 (-20)
SQ 30 FD 50 SQ 30
FD 30 RT 90
REPEAT 2 [FD 30 SQ 30]
FD 30 RT 90 FD 50 SQ 30
FD 30 RT 90 FD 60
STAIRS
END
```



```
TO STAIRS
REPEAT 2 [RT 90 FD 6 RT 90 FD 30 LT 90 FD 6 LT 90 FD 30]
END

TO SQ  :N
REPEAT 4 [FD  :N RT 90]
END
```

```
TO CHOOSE
PRINT [UPSTAIRS THERE ARE FIVE ROOMS.]
PRINT [ONE CONTAINS TREASURE.]
WAIT 200
PRINT [YOU HAVE TIME TO OPEN ONLY TWO DOORS.]
PRINT [CHOOSE 1 2 3 4 OR 5.]
TEST READWORD = 3
IFT PRINT [YOU FOUND THE TREASURE!] WAIT 100 TREASURE
IFF PRINT [SORRY! THAT ROOM'S FULL OF TRASH.] TRY.AGAIN
END
```

(Note: When READWORD is used with a numeral, you do not need the quotation mark before the numeral.)

And a procedure to allow one more choice:

```
TO TRY.AGAIN
PRINT [YOU HAVE ONE MORE CHANCE.]
PRINT [CHOOSE 1 2 3 4 OR 5.]
TEST READWORD = 3
IFT PRINT [YOU FOUND THE TREASURE!] WAIT 100 TREASURE
IFF PRINT [SORRY! MISSED AGAIN.] WAIT 100 OUT
END
```

The TREASURE can be whatever you want it to be. How about designing a pattern with flashing tiles for DIAMONDS?

```
TO TREASURE
CS
PRINT [IT'S A ROOM FULL OF DIAMONDS!]
PRINT [STUFF YOUR POCKETS AND GET OUT!]
DIAMONDS
WAIT 200
OUT
END
```

We'll define one more procedure to get our adventurers OUT of the castle.

```
TO OUT
CS NOTURTLE
PRINT [YOU JUMP THROUGH THE WINDOW]
PRINT [AND RACE FOR YOUR HORSE.]
PRINT [ ]
PRINT [WHAT HAPPENS NEXT?]
PRINT [IT'S UP TO YOU!]
PRINT [ ]
PRINT [IN OTHER WORDS, YOU'RE ON YOUR OWN NOW!]
END
```

And so, Fair Adventurer, you *are* on your own. Aristotle and Little Bit have given you a start. It's your adventure now. How are you going to improve it? What choices are you going to add? What graphics will you design? Where are you going from here?

Appendix

Tile Number and Characters

Group 1	
Number	*Character*
32	(space)
33	!
34	”
35	#
36	$
37	%
38	&
39	‘

Group2	
Number	*Character*
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/

Group 3	
Number	*Character*
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7

Group 4	
Number	*Character*
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?

Group 5	
Number	*Character*
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G

Group 6	
Number	*Character*
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O

Group 7	
Number	*Character*
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W

Group 8	
Number	*Character*
88	X
89	Y
90	Z
91	[
92	\
93	]
94	^
95	_

Colors

Clear	0	Rust	8
Black	1	Orange	9
Green	2	Yellow	10
Lime	3	Lemon	11
Blue	4	Olive	12
Sky	5	Purple	13
Red	6	Gray	14
Cyan	7	White	15

Chapter 2

Optical Triangle

```
TO OP.TRI
REPEAT 3 [FD 15 LT 120]
REPEAT 3 [LEG FD 15 LT 120]
END

TO LEG
FD 20 LT 120 FD 30
RT 120 FD 5 RT 60 FD 35
RT 120 FD 30 RT 120 FD 5 RT 60
END

TO SWIRL.OP.TRI
PU FD 20 PD
REPEAT 6 [OP.TRI PU BK 10 LT 120 FD 10 RT 60 PD]
END
```

```
TO AMULET
SUN
INSIDE.CIR
PU HOME PD
CIRCLE.SUN
CIR2
CENTER
REPEAT 6 [TRI 10 RT 60]
END

TO SUN
REPEAT 18 [POINT CONNECT]
END

TO INSIDE.CIR
PU BK 3 RT 75 PD
REPEAT 18 [FD 4 RT 20]
END

TO CIRCLE.SUN
FD 15 RT 90
REPEAT 18 [FD 10 RT 20]
END

TO CIR2
PU LT 90 FD 3 RT 90 BK 1 PD
REPEAT 18 [FD 11 RT 20]
END

TO CENTER
PU HOME RT 160 FD 15
SH 0 PD
END

TO TRI  :N
REPEAT 3 [FD  :N RT 120]
END

TO POINT
FD 15 RT 160 FD 15 RT 100 FD 5 RT 100
END

TO CONNECT
LT 100 BK 5 RT 120
END
```

Optical Hexagon

```
TO OPTICAL.HEXAGON
TRIPLET1
TRIPLET2
END

TO TRIPLET1
S
REPEAT 2 [PU FD 10 PD S]
END

TO S
SIDE LT 60 DIA
RT 60 FD 5 LT 60
VEE VEE2 VEE3
SIDE2
END

TO SIDE
REPEAT 2 [FD 5 RT 60 FD 15 RT 120]
END

TO DIA
REPEAT 2 [FD 5 RT 60 FD 5 RT 120]
END

TO VEE
FD 5 RT 120 FD 10 LT 120 FD 10 RT 120
FD 5 RT 60 FD 15 RT 120 FD 15
END

TO VEE2
RT 60 FD 5 RT 120 FD 10 RT 180
FD 5 RT 60 FD 10 RT 60 FD 15 RT 120
FD 5 RT 60 FD 10 LT 60 FD 10 RT 120
END

TO VEE3
RT 60 FD 10 RT 60
FD 10 RT 60 FD 15 RT 120 FD 5 RT 60
FD 10 LT 60 FD 5 RT 60 FD 5
END

TO SIDE2
RT 120 FD 10 LT 60 FD 5 RT 120
FD 15 RT 60 FD 5
END
```

```
TO TRIPLET2
S2
REPEAT 2 [PU BK 15 LT 60 BK 15 LT 60 PD S2]
END

TO S2
L L2 L3
BK 5 RT 120 FD 10 LT 120 DIA
END

TO L
FD 5 LT 60 FD 10 RT 60 FD 10 LT 60
FD 5 LT 120 FD 15 LT 60 FD 15
END

TO L2
LT 60 FD 15 RT 120 FD 10 RT 120 FD 5
RT 60 FD 5 RT 60 FD 5 RT 120 FD 10
LT 120 FD 5 LT 60 FD 15 LT 120 FD 20
END

TO L3
BK 20 RT 60 FD 5
LT 60 FD 15 RT 60
FD 5 LT 60 FD 5
END
```

Editing Features for TI LOGO

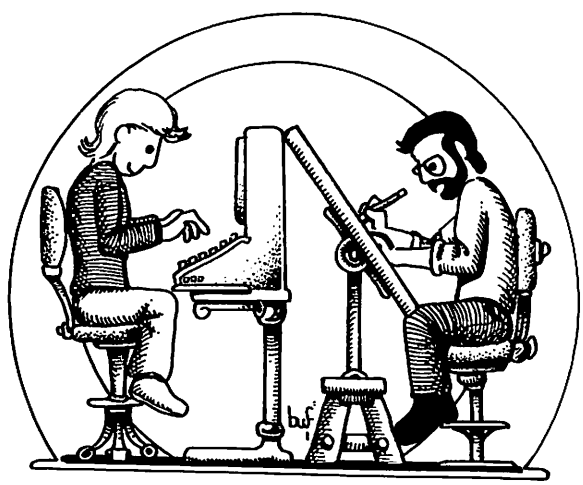
TI 99/4

SHIFT W	Moves cursor to beginning of line
SHIFT V	Moves cursor to end of line
SHIFT ↑	Moves cursor up one line
SHIFT ↓	Moves cursor down one line
SHIFT ←	Moves cursor one space to left
SHIFT →	Moves cursor one space to right
ENTER	If cursor is at the end of a line, opens a space for a new line. Otherwise moves the cursor, the character immediately above it and everything to the right down to the next line.
SHIFT T	Erases the character or space one space to the left of the cursor. If the cursor is under the first character of a line, moves line up one line.
SHIFT F	Erases the character or space immediately above the cursor. If the cursor is at the end of a line, moves next line up.
SHIFT C	Erases the character or space above the cursor and everything to its right.
SHIFT Z	Leaves the Edit Mode

TI 99/4A

FNCT 5	Moves cursor to beginning of line
FNCT 6	Moves cursor to end of line
FNCT ↑	Moves cursor up one line
FNCT ↓	Moves cursor down one line
FNCT ←	Moves cursor one space to left
FNCT →	Moves cursor one space to right
ENTER	If cursor is at the end of a line, opens a space for a new line. Otherwise moves the cursor, the character immediately above it and everything to the right down to the next line.
FNCT 3	Erases the character or space one space to the left of the cursor. If the cursor is under the first character of a line, moves line up one line.
FNCT 1	Erases the character or space immediately above the cursor. If the cursor is at the end of a line, moves next line up.
FNCT 4	Erases the character or space above the cursor and everything to its right.
FNCT 9	Leaves the Edit Mode.

DONNA BEARDEN'S first big writing break came in sixth grade when she translated a well-known fairy tale into a play with all the characters talking in verse. Her teacher was so impressed, she let Donna cast, direct and present the play for the fifth grade class. There followed a long dry spell until eighth grade when she won second place, three pairs of shoes, in a local radio station's essay contest on "Why I want to go back to school." (She entered under a pseudonym so none of her friends would know.) After several years of writing in the fields of aging and health, she was dragged reluctantly into the computer world when she was asked to write an article on computer adaptations for the handicapped. What was a dreaded assignment turned into love at first byte. The rest is history. This is her third book on LOGO for Reston.



BRAD W. FOSTER was born with a pen in his hand, an event causing no end of discomfort for his mother. It took several decades before he found out what the pointed thing was for. After an embarrassingly short career as a professional darts player, someone stopped laughing long enough to tell him about ink. The rest, as they say, is a tiny footnote to history. Since that time he has appeared in well over two semi-professional publications, not to mention... Well, let's not mention it. Although he has almost negative knowledge of computers, he managed to cover up that deficiency by not drawing anything even remotely resembling a computer in any of the dozens of illustrations in this book. It is this bizarre talent for taking the completely novel approach to his subject that explains the heights to which his artistic career has soared. His next major project is illustrating a children's book, provided he can learn how to read the story and find out what to draw — or not to draw, in his case. His motto is "leave no piece of paper blank."

$10.95

Donna Bearden

A Bit of Logo Magic

Adventures for Intermediate Programmers

The author of "1, 2, 3, My Computer and Me: A Logo Funbook for Kids" has done it again, only this time for kids and kids-at-heart who understand the basic fundamentals of Logo and are ready to learn more.

Join Aristotle, the wise old wizard, and Little Bit, a mischievous dragon, as they travel from one magical adventure to another—adventures of visual magic such as optical illusions and tessellations, and of word magic such as vocabulary quizzes, funny cliches, and a do-it-yourself adventure game.

Contents

A Word
A Little Bit of Mischief
Elaborate Designs with Simple Shapes
Patterns, Tessellations, and Optical Illusions
Spider Webs and Other Magnificent Designs
Questions, Quizzes, and Quotations
Fractured Fables and Customized Cliches
An Adventure in the Dark Forest

This is Donna Bearden's third Logo book with Reston Publishing Company. Brad Foster, artist extraordinare, has performed his own remarkable magic in the book's illustrations.

The procedures in this book all run on TI99/4A Logo.

RESTON PUBLISHING COMPANY, INC.
A RESTON COMPUTER GROUP BOOK
A Prentice-Hall Company
Reston, Virginia 22090

Cover design by Brad Foster

0-8359-0494-6

TI-99/4A is a registered trademark of Texas Instruments